

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2001-306308
(P2001-306308A)

(43)公開日 平成13年11月2日(2001.11.2)

(51)Int.Cl.	識別記号	F I	テーマコード(参考)
G 0 6 F 9/06	5 3 0	G 0 6 F 9/06	5 3 0 P 5 B 0 7 6
9/45		9/44	5 3 0 P 5 B 0 8 1
9/44	5 3 0		3 2 0 C

審査請求 未請求 請求項の数7 O L 外国語出願 (全 73 頁)

(21)出願番号 特願2000-109978(P2000-109978)

(22)出願日 平成12年4月11日(2000.4.11)

(71)出願人 300015447

エスアーベー アクチエンゲゼルシャフト
ドイツ連邦共和国 ヴァルドルフ ノイロ
ットシュトラッセ 16

(72)発明者 デイヴィッド ホーリー

千葉県松戸市稔台72-7ハーヴェスト・ヒ
ル503

(74)代理人 100061815

弁理士 矢野 敏雄 (外3名)

Fターム(参考) 5B076 DA01 DC01

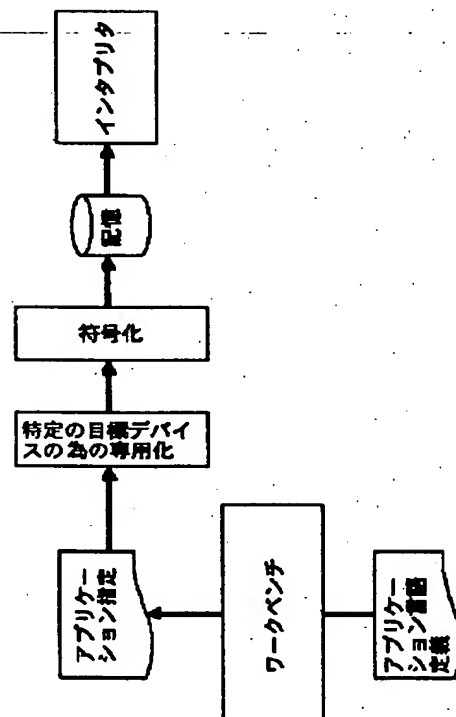
5B081 AA10 DD01

(54)【発明の名称】 データ中心アプリケーションのクラス定義方法

(57)【要約】 (修正有)

【課題】 多くの異なる種類のコンピューティングデバイス上で使用できるデータ中心アプリケーションを短時間で作成するために、共通のユーザ・インタフェースモデルを有するアプリケーションのクラスを定義する枠組みを定義する。

【解決手段】 アプリケーションシステムは、ユーザ・インタフェースモデル内でアプリケーションを表現するために必要な言語エレメントを定義するアプリケーション指定言語、ワークベンチ、及びインタプリタを有する。アプリケーション開発者はワークベンチと呼ばれるコンポーネントを使用して、言語エレメントのインスタンスを作成、組立し、アプリケーション指定を形成する。インタプリタは、アプリケーション指定を実行し、コンピューティングデバイス上にユーザ・インタフェースを提示する。



BEST AVAILABLE COPY

【特許請求の範囲】

【請求項1】 特定の総称的ユーザ・インタフェースモデルを共有するデータ中心アプリケーションのクラスを定義する方法において、以下のステップ：前記総称的ユーザ・インタフェースモデルが必要とする、タイルと呼ばれる特別ユーザ・インタフェースエレメントと、それらの構成エレメントのタイプを決定し；アセンブリと呼ばれるタイル群内の、タイル間の濃度と関係を決定し；アセンブリ間のナビゲーション用の論理モデル、及びモデル状態上で必要なユーザ操作を決定するステップを有したことを特徴とする方法。

【請求項2】 アプリケーションの前記クラスのそれぞれに適したメタ言語によりアプリケーション指定言語を定義する方法において、以下のステップ：タイルタイプと構成エレメントを指定するエレメント、エレメント間の包含関係、及びエレメント間の参照関係を表すために必要な属性を定義し；タイルとユーザ・インタフェースエレメントの特性を表すために必要とされるエレメントの属性を定義し；ナビゲーションモデルにより必要とされるエレメントの属性を定義するステップを有することを特徴とする方法。

【請求項3】 アプリケーションの前記クラスのそれぞれに対するアプリケーションシステムを、前記クラスのインスタンスの開発と実行のために作成する方法において、総称的ワークベンチを、メタ言語で表現不可能なアプリケーション指定の妥当性に関する制約を実現するイベントハンドラでカスタマイズし；総称的インタプリタをカスタマイズするステップを有し、このインタプリタカスタマイズは以下のステップ：

- a) タイル、ユーザ・インタフェースエレメント、及びアセンブリの指定に対するオブジェクトモデルを作成し；
- b) アプリケーション指定又はそのフラグメントからオブジェクトモデルインスタンスを作成する定義オブジェクトコンストラクタを実行し；
- c) 定義オブジェクトから、適切なコンピュータプログラミング言語でのUIモデルインスタンスの作成を実行するステップを有することを特徴とする方法。

【請求項4】 特定のクラスのアプリケーションを作成する方法において、前記アプリケーションの指定を含む文書を、アプリケーションシステムのワークベンチコンポーネントを使用して作成し；オプションとして、前記アプリケーション指定文書を特定のデバイス用に専用化し；アプリケーション指定文書を、インタプリタに適した形式に符号化するステップを有することを特徴とする方法。

【請求項5】 前記総称的ユーザ・インタフェースモデルのそれぞれに対応する複数の具体的なユーザ・インタフェースを提供する方法において、

UIモデルの個々のエレメントに対して、デバイスの出力媒体上の表示を定義し；UIモデル上の個々の操作に対する具体的なユーザ・インタフェースを定義し；出力表示とUIモデル操作ユーザ・インタフェースの必要に応じて、アプリケーション指定言語内に付加的なエレメント及び属性を定義し；個々のテーマを適切なコンピュータ言語でインタプリタのコンポーネントとして実行し；ユーザ・インタフェースが提示される特定のコンピュータデバイスに適した適切なテーマをインタプリタが選択できるようにする機構を提供するステップを有したことを特徴とする方法。

【請求項6】 ユーザに対して、遠隔システムとの対話を外見上、反応-応答形式で提示することにより、モバイルデバイス上で前記アプリケーションのオンライン、オフライン、及びそれらの混合動作を提供する方法。

【請求項7】 特定の環境のために、様々なインタプリタを作成する方法。

【発明の詳細な説明】**【0001】 1. 発明の分野**

本発明はコンピュータデバイス用のデータ中心アプリケーションの作成に関し、より詳細には共通のユーザ・インタフェースモデルを共有するアプリケーションのクラスを定義するための方法、異なるユーザ・インタフェース能力、寸法、及び形状を有した様々なコンピュータデバイスに適した態様でユーザ・インタフェースモデルを提示する技術、及び、断続的に接続されるオンラインモード、又はオフラインモードで使用可能な順応型アプリケーション(conformant applications)を短時間で作成する技術に関する。

【0002】 2. 発明の背景**2. 1 問題点**

近年の技術の進展の結果、必ずしも、或いは全くデスクトップコンピュータを使用できない環境における専門職又は非専門職に従事する労働者に対して、低廉でメンテナンスをあまり要しない小型のモバイル(“普及型”；“pervasive”)デバイスを提供することが可能になった。このような普及型デバイスは企業データシステム及びITを利用したビジネスプロセスと統合可能でなければならない。さらに、現在の通信信頼性、通信コスト、及びデバイスのバッテリー容量を考慮すると、接続モードのみならず非接続モードをサポートする必要がある。残念ながら、これら2つの条件を満たすツールは不足している。

【0003】 アプリケーションの開発、保守、及び実装の観点からすると、複数のデバイス画面寸法(device form factors)に対応しなければならないという付加的な問題がある。例えば、ウィンドウズCE一般消費者向けデバイスは少なくとも4種類の異なるスクリーンサイズを有する。マイクロブラウザを搭載したワイヤレス電話も非常に大きな関心を集めている。最後に、最も普及し

ているデバイスは昔ながらの電話である。音声媒体もまた、企業コンテンツを、インタラクティブ・ボイス・レスポンスシステムを介して、電話のプッシュボタンや音声認識／合成インタフェースを使用して伝達可能である。開発者はこのような様々な、次々と誕生するデバイスに対するソリューションを提供することがますます困難になってきている。

【0004】2. 2課題

普及型コンピューティングデバイス用のカスタマイズ可能なアプリケーションを短時間で作成するための方法を提供する。アプリケーションはSAP R/3等の企業データシステムと統合可能でなければならない。さらに、通信インフラストラクチャは急速にかつ、世界中で不統一に発展しているため、開発方法は、開発されたアプリケーションのオンライン及びオフライン双方での使用において過大な非連続性をもたらしたり、その結果、エンドユーザにおいて過大な訓練費用が必要となることのないようなものでなければならない。

【0005】上記デバイスにおけるアプリケーションロジックに対する要求は通常低い。それは、このようなデバイス上で実行されるビジネスプロセスは比較的単純であり、また、デバイス自体が制限された能力しか有していないからである。

【0006】3. 発明の要旨

本発明は、多くの異なる種類のコンピューティングデバイス上で使用できるデータ中心アプリケーションを短時間で作成するための方法に関する。

【0007】アプリケーションは高水準で指定され、オブションとして特定の媒体又は画面寸法用に特化され、適切なインタプリタにより実行される。この目的で、宣言型アプリケーション指定言語のファミリーが指定される。これらの言語は、データモデルの表現、企業データとの統合、及び、抽象化されたユーザ・インタフェースが可能である。それぞれの言語は特定の総称的ユーザ・インタフェースパラダイムに対応し、パラダイムに適合するアプリケーションを指定するために使用される。ユーザ・インタフェースパラダイムは、個々のデバイスおよび媒体の能力に見合った方法で、様々なデバイスへマッピングされる。個々の言語用のインタプリタはアプリケーション指定を翻訳し、ユーザ・インタフェースを提示する。インタプリタは基本的なユーザ・インタフェースの枠組み（ナビゲーションモデル、データのUIエレメントへの結合）、データサービス（妥当性検査、オブジェクト間関係とナビゲーション、データイベント制御、データ変更の取り込み）、及びミドルウェアサービス（データ同期、仮想要求-応答）を提供する。内蔵機能は、共通のスクリプト言語の使用、及び、COM又は同様の技術に基づくカスタムオブジェクトコンポーネントの使用により、強化することができる。クライアントデバイスとサーバ間の統合は、インタプリタに内蔵された

データ同期機構により実現される。従って、動作の基本モードはオフラインである。しかし、オフラインとオンライン双方でのアプリケーションの混合使用も、インタプリタの付加的な能力によりサポートされている。この付加的な能力とは、ユーザがオンライン接続において預期する“要求-応答”型のインタラクションをシミュレートする能力である。インタプリタはこのような見かけ上の能力を、以下のことを実現することにより提供する。すなわち、1) データ同期がほぼ実時間で行われ、2) 同期から生じるデータ変更が、結果が利用可能となった時に明示的なユーザによる操作を要することなくユーザ・インタフェースに反映される事、である。前者の条件はコンピューティングデバイスの標準的なメッセージ及び通信サービスにより実現可能である。後者の条件を満たすためには、データ同期層がユーザ・インタフェースデータイベント層への適切なフックを有するようにインタプリタを構成することにより、サーバ側で発生したデータ変更が自動的にUIエレメントに送られるようにする。

【0008】本発明の課題は、共通のユーザ・インタフェースモデルを有するアプリケーションのクラスを定義する枠組みを定義することである。

【0009】本発明の別の課題は、個々の特定のユーザ・インタフェースモデル用のアプリケーション開発システムを作成するためのプロセスと方法を定義することである。

【0010】本発明の別の課題は、ユーザ・インタフェースモデルの特定のアプリケーションを指定するために使用するコンピュータ言語を定義する方法を提供することである。

【0011】本発明の別の課題は、特定のユーザ・インタフェースモデルを実現する、特定のコンピューティングデバイス用のインタプリタを提供することである。

【0012】本発明の別の課題は、個々のコンピューティングデバイスに適合した態様で特定のユーザ・インタフェースモデルを実現するレイアウトコンポーネントをインタプリタ内に提供することである。

【0013】本発明の別の課題は、同一のユーザ・インタフェースモデルと実現を使用して、アプリケーションのオフライン、オンライン、及びそれらの混合使用形態を可能とすることである。

【0014】本発明の別の課題は、デバイスと1つ又は複数のサーバ間でデータを同期するための機能性を、特定のコンピューティングデバイスのインタプリタ内に提供することである。

【0015】本発明の別の課題は、サーバ側で生じたデータ変更を明示的なユーザによる操作を要することなくユーザ・インタフェース内に反映するための機能性を、特定のコンピューティングデバイスのインタプリタ内に提供することである。

【0016】本発明の別の課題は、パーソナルデジタルアシスタント(Personal Digital Assistant)等の小型コンピュータデバイス上の前記インタプリタの性能を向上させるための技術を提供することである。

【0017】本発明の別の課題は、特定のユーザ・インタフェースモデルに適合されたアプリケーションを作成するためのプロセスと方法を定義することである。

【0018】4. 公知技術

4.1 複数のデバイス画面寸法及び媒体をサポートするための、サーバに基づくレイアウト

請求項(a)に関連

これらのソリューションは、オンライン使用を対象とする。場合によっては、これらのソリューションを、オフラインブラウジングの使用により、限定されたオフラインシナリオに応用することも可能である。後者の技術は、提示(presentation)における柔軟性(データの動的選択及びフォーマット)、データ妥当性検査、及びデータボリュームスケーラビリティ(データ同期のサポート等)に関連する多くの周知の短所を有する。

【0019】4.2 フォーム言語

請求項(c)に関連

これらの言語は、ユーザ・インタフェースのみを対象とする。一例はCelesta SmartFormsである。フォーム言語は単純なデータモデルしかサポートできず、企業データとの統合性も貧弱である。

【0020】4.3 音声プロキシ

請求項(a)に関連

これらのゲートウェイサービスは、音声合成と音声認識により、既に存在するウェブページを動的に、電話に適した形式に変換しようとするものである。これらのプロキシは、大量の視覚情報を狭帯域音声インタフェース上に提示するために、推論に基づいてウェブページ上に構造を配置しなければならない。

【0021】公知技術では、このアプローチは適応範囲及び能力の点で限界がある。

【0022】5. アプリケーションモデル

5.1 アプリケーションのモデル化

アプリケーションは、いくつかの関連したコンポーネントの組み合わせであり、以下のモデルに従う：

アプリケーション=データモデル+ミドルウェアモデル+ビジネスロジック+ユーザ・インタフェース

ここで定義するアプリケーション指定言語(Application Specification Languages)により、上記コンポーネントを個々にアドレスすることが可能となる。ただし、上記したコンポーネントの全てが必要なわけではない。データモデルはアプリケーションが動作する基礎となるローカルデータ、即ちテーブルの集合とそれらの相互関係を定義する。ミドルウェアモデルは、クライアントデバイスとサーバ間で交換されるメッセージの内容を定義してデータ同期を取るとともに、メッセージとデータモデ

ルのオブジェクト間の関係を定義する。ビジネスロジックコンポーネントはファンクションのプールと、様々なアプリケーションイベントに関連付けられたスクリプトから構成される。ユーザ・インタフェースは、ユーザがアプリケーションと対話するための手段を定義する。

【0023】さらにユーザ・インタフェースコンポーネントを以下のような要素に分解する：

ユーザ・インタフェース=ユーザ・インタフェースモデル+テーマ

ユーザ・インタフェースモデル=タイル+アセンブリ+ナビゲーション

タイル=カスタム、コンタクト等の、アプリケーションレベルオブジェクトのビュー。

【0024】提示されたインタフェースエレメントの集合。

【0025】アセンブリ=タイルの群とそれらの相互関係

例：関連ビジネスオブジェクトの集合に対するビューの一群。

【0026】ナビゲーション=ユーザ・インタフェースモデルの状態モデル、並びに、タイルとアセンブリ間のアクセス、インスタンス化、及びトランジションを含む、UIモデル上の操作。ナビゲーション状態は以前の状態の記憶も含むことができる。

【0027】テーマ=特定のデバイス形式上でのユーザ・インタフェースモデルの提示。テーマの実現には、ランタイムにおける提示のステータスを表す付加的“状態”情報が必要な場合もある。この情報は論理的にナビゲーション状態に関連付けられ、それを拡張する。

【0028】上記の各等式は、タイル、アセンブリ、及びナビゲーションモデルの具体的定義によりパラメタ化されたアプリケーションのファミリを定義する。選択される特定のタイル、アセンブリ、及びナビゲーションモデルから、対応UIモデルに適合するアプリケーションを特定するアプリケーション特定言語の必要条件が分かる。結果として得られるアプリケーションは、個々の特定のデバイス及び媒体の種類に適したテーマに従って、様々なデバイス及び媒体上に提示される。以下では、幾つかのユーザ・インタフェースモデルを提示する。

【0029】5.2 MDF (タイルセット・スタックモデル)

MDFフレームワーク用のユーザ・インタフェースモデルは、全てのアプリケーション特定言語の場合と同様、タイル、アセンブリ、及びナビゲーションモデルの定義により特徴付けられる。MDFに対しては、タイル=特定のビジネスオブジェクトタイプ内のフィールドに結合されたUIエレメントの集合。例えば、カスタム詳細タイルはカスタムのコレクションに結合され、個々のUIエレメントはカスタム番号、氏名、業種等のフィールドに結合される。

【0030】アセンブリ＝階層（又は階層のコレクション）内に組織された、タイルセットと呼ばれるタイルの集合。1つ又は複数のタイルは“メイン”タイルと呼ばれ、それぞれ“サブ”タイルのツリーのルートを構成する。サブタイルは、それに対応付けられた全てのデータアイテムの集合上の照会(query)により、そのペアレントに関連付けられる。この照会はペアレント内で現在選択されているアイテムの特性によりパラメタ化されている。これにより、例えば、メインタイル内のあるカスタマを選択することにより、サブタイル内の関連性を有する売り注文を自動的に選択・表示可能となる。

【0031】ナビゲーション＝ナビゲーション状態はアセンブリ（タイルセット等）インスタンスのスタックを有し、その最上位がカレント(current)であるとされる。カレントタイルセットのみが表示され、ユーザと対話可能である。論理的には、全てのタイルが同時にユーザに提示されるが、特定のデバイスにおけるテーマレイアウトによっては、ユーザは明示的に何らかの動作を行って個別のタイルにアクセスすることが必要となる場合もある。そのような動作はナビゲーションスタックに影響を与えない。新規のタイルセットインスタンスは、いずれかのタイル内で“ハイパーリンク”を実行することにより作成可能であり、その場合、新規インスタンスはスタックの最上位に置かれる。この動作は、スタックの最上位を廃棄することにより、逆転できる。いくつかのタイルセットは最上位タイルセットとして区別される。そのようなタイルセットを直接作成するための機構があり、その場合スタックは、新規インスタンスの作成前にクリアされる。

【0032】このユーザ・インタフェースモデル、及び対応するアプリケーション言語は好適な実施例である。このモデル用のアプリケーション指定言語の表1を参照されたい。MDFテーマの例は第8. 3. 1章に記載されている。

【0033】5. 3 普及型ワークフロー（ダイアログモデル）

“ダイアログ”UIモデルは単純なワークフロー(Workflow)アプリケーションに適しており、エンドユーザは、ワークフロー内の特定の時点における少量の所定のデータに基づいて単純な選択を行うよう要求される。この選択ステップはモバイルコンピューティング・デバイス上で提示可能である。このシナリオに適したユーザ・インタフェースモデルは以下の定義により記述される：
タイル＝ワークフローコンテナ上のフィールドに結合されたUIエレメントの集合。例えば、休暇請求承認用のワークフロー・コンテナは、請求者、日付、理由その他のフィールドに結合されたUIエレメントを有する。

【0034】アセンブリ＝アセンブリは1つのタイルを有する。理解を容易にするために、この章内の以下の説明では、タイルと、それが構成するアセンブリの間に区

別を設けないこととする。

【0035】ナビゲーション＝ナビゲーション状態は、カレントタイルと呼ばれる一つのタイル・インスタンスを有する。ユーザはこのタイルとのみ対話可能である。全てのタイルインスタンスはプール内で利用可能であり、ユーザは該プールにアクセスして任意のタイルインスタンスをカレントタイルインスタンスと命名することができる。ユーザはプールから別のインスタンスを選択することにより、任意のインスタンスをプールに戻すことができる。タイルインスタンスの処理が完了したことを示すためにユーザが利用できる機構が設けられており、処理終了後、タイルはプールから論理的に取り除かれる。ユーザはまた、アプリケーション内に定義されたタイルのリストから選択して新規タイルインスタンスを開始し、プールに付加することができる。

【0036】このUIモデルは構成を複雑化することにより、より複雑なタイルインスタンスプールを管理可能である。例えば、個々のタイルインスタンスにより表されたワークフローは、対応優先順位又は付加的状态を有することができる。

【0037】6. アプリケーションシステム

6. 1 概観

アプリケーションシステムは、特定のユーザ・インタフェースモデルを共有するアプリケーションを作成するための機構を提供する。

【0038】アプリケーションシステムはアプリケーション指定言語、ワークベンチ、及びインタプリタを有する。アプリケーション指定言語は、ユーザ・インタフェースモデル内でアプリケーションを表現するために必要な言語エレメントを定義する。アプリケーション開発者はワークベンチと呼ばれるコンポーネントを使用して、言語エレメントのインスタンスを作成・組立し、アプリケーション指定を形成する。インタプリタは、アプリケーション指定を実行し、コンピューティングデバイス上にユーザ・インタフェースを提示するコンポーネントである。

【0039】アプリケーションシステムを作成するステップは以下のステップを有する。

【0040】第1のステップはユーザ・インタフェースモデルを定義するステップであり、以下のステップを有する：タイル、及びタイル上に含まれる全てのユーザ・インタフェースのタイプと機能を判断し；アセンブリ内のタイル間の濃度と関係、及び特定の専用化アセンブリ内のタイル間の操作に対する特定の要求を判断し；論理ナビゲーションモデルと、モデル状態上の要求されたユーザ動作を判断する。

【0041】第2ステップはメタ言語によりアプリケーション指定言語を定義し、以下のステップから構成される：タイル形式（タイプ）、タイル内に（あるとすれば）含まれるユーザ・インタフェースエレメント、エレ

メント間の包含関係、及びエレメント間の参照関係を表すために必要とされる属性を指定するエレメントを定義し；タイトルとユーザ・インタフェースエレメントの特性を表すために必要とされるエレメント上の属性を定義し；ナビゲーションモデルにより必要とされるエレメント上の属性を定義する。

【0042】第3ステップは総称的ワークベンチをカスタマイズするステップであり、以下のステップから構成される：メタ言語で表現不可能なアプリケーション指定の妥当性に関する制約を実現するイベントハンドラを作成する。

【0043】第4ステップでは個々のサポートされたデバイスのためのレイアウトテーマが定義され、以下のステップを有する：UIモデルの個々のエレメントに対して、出力媒体上の表示を定義し；UIモデル上の個々の動作に対するユーザ・インタフェースを定義し；UIモデル上の個々の動作のユーザ・インタフェースを定義し；出力表示及びUIモデル動作ユーザ・インタフェースにより必要とされる付加的エレメント及び属性をアプリケーション指定言語内に定義する。

【0044】第5ステップはインタプリタのUIモデルを実現し、以下のステップ：タイトル、ユーザ・インタフェースエレメント、及びアセンブリの各指定のオブジェクトモデルを作成し、該オブジェクトを“定義オブジェクト”と呼び；アプリケーション指定又はそのフラグメント(fragments)からオブジェクトモデルインスタンスを作成する定義オブジェクトコンストラクタ(construct or)を実行し；適切なコンピュータプログラミング言語で、定義オブジェクトからUIモデルインスタンスの作成を実行するステップを有する。

【0045】第6ステップはインタプリタ内にテーマを実現するステップであり、個々のテーマに対して以下のステップを有する：適切なコンピュータ言語でそれぞれのテーマをレイアウトハンドラとして実行し；インタプリタによるコンピューティングデバイスに対する適切なテーマ選択を可能にする機構を提供する。

【0046】このプロセスの詳細は第8章に、好適な実施例すなわちMDFと共に記載されている。

【0047】アプリケーション指定言語の構文の好適な実施例はXML(eXtensible MarkupLanguage; World Wide Web Consortium, www.w3c.org)である。この文脈におけるこの“XML”という用語の使用はWorld Wide Web ConsortiumによるXMLの指定、及びXMLをバイナリ又はその他のXML構造を保存する形式に符号化するスキームを指す。

【0048】前記構文の別の好適な実施例はXMLのバイナリ符号化である、“WAPバイナリXML符号化”(WAP Binary XML encoding; WAP Consortium, www.wapforum.org.)である。

【0049】アプリケーション指定言語を定義するため

のメタ言語の好適な実施例はXMLスキーマ(Schema)(World Wide Web Consortium, www.w3c.org)である。

【0050】6.2 ワークベンチ

ワークベンチはアプリケーション開発者がアプリケーションを作成するための手段である。その動作はアプリケーション指定言語の定義を含む文書により制御される。ワークベンチは前記言語に従うために、その文書ブラウジング及び編集機能を拘束する。前記に関わらず、アプリケーション指定言語を定義したメタ言語により適切に表現されなかった文書妥当性基準を実現するために、ワークベンチに何らかのカスタマイズが必要な場合もある。

【0051】6.2.1 アーキテクチャ

図3は、ワークベンチの単純化オブジェクトモデルを示す。ワークベンチはタイプされた文書(Typed Document)及び対応する文書カーソル、タイプされた文書をブラウジング及び編集するためのツリービュー(TreeView)及びエレメント属性パネル(Element Attribute Panel)、及び開発プロセス中に文書の最終表示をシミュレートするための多くのプレビューア(Previewers)を含む。タイプされた文書コンポーネントはアプリケーション指定言語定義を指定して指定文書構造を理解し、個々のエレメント及び属性上で許容される編集動作を決定する。

【0052】タイプされた文書コンポーネントはアプリケーションプログラミングインタフェース(API)を提供する。APIは以下の目的で使用される。すなわち、1)アプリケーション指定言語のロード、2)アプリケーション指定文書の作成、ロード及びセーブ、及び3)その他のコンポーネントに対して有効編集オペレーションに関する情報を供給することによる、該コンポーネントの適切な動作の誘導。アプリケーション指定文書に対する変更は全て、その他全てのコンポーネントに同報通信(broadcast)され、それらコンポーネントはそれぞれ内部状態を更新し、該更新に従った表示を行う。

【0053】“カレントエレメント”という考え方は、文書カーソル(Document Cursor)により実現される。このコンポーネントは、カレントエレメントと呼ばれるアプリケーション文書内の、特別なエレメントに対する参照を含む。文書エレメントをブラウズ可能なコンポーネント、例えばツリービューやプレビューアはいずれも、文書カーソルAPIを呼び出すことにより、カレントエレメントレファレンスを更新できる。文書カーソルは、新規レファレンスを全ての他のコンポーネントに連絡し、それら他のコンポーネントはそれぞれ内部状態を更新し、該更新に従った表示を行う。

【0054】ツリービューは、文書の実際の構造を反映した階層として文書を表示する。コンテキスト対応(context-sensitive)メニューは、文書に対するエレメントの削除又は挿入のために設けられる。

【0055】エレメント属性パネルは、カレントエレメ

ント用に定義された属性の集合を表示する。属性はパネル上で、それらのタイプ及びその他のメタ属性に適した形で表される。

【0056】プレビューアコンポーネントはアプリケーション文書を、インタプリタにより実行された場合の提示態様に近似した形式で提示する。個々のプレビューアコンポーネントはそれぞれ異なる具体的提示（テーマ）を実現し、ワークベンチユーザ・インタフェースを介して個別に開始することができる。

【0057】カレントエレメントは、ユーザ・インタフェースエレメントに対応する場合は、個々のオープンプレビューア内で見ることができ、ハイライト表示される。また、プレビューア内でユーザ・インタフェースエレメントの表示を選択することにより、当該エレメントがカレントエレメントになる。その結果、ツリービュー及び全てのプレビューアはそれぞれの表示を適切に更新する。

【0058】6. 2. 2 動作

ワークベンチの動作の詳細は第7. 2章に説明してある。

【0059】6. 3 インタプリタ

インタプリタは、アプリケーション指定を実行し、ユーザ・インタフェースを提示するコンポーネントである。

【0060】6. 3. 1 アーキテクチャ

インタプリタのアーキテクチャは図7に示されている。特に重要なのは、“XMLサービス”と題されたサブシステムである。このサブシステムは、左端に例示としてGUI定義、データモデル、及びミドルウェアモデルコンポーネントに分けられたアプリケーション指定に対するアクセスを提供する。右端には、デバイス・ネイティブ・UIサービスがある。これらは、特定のコンピューティングデバイスにより提供されるユーザ・インタフェースサービスであり、グラフィカルディスプレイを有するデバイス用のウィンドウ管理、音声インタフェースを有するデバイス用の音声合成等を含む。その他のコンポーネントは集合的にランタイムコンポーネントと呼ばれる。

【0061】・タイルハンドラコンポーネントはタイルタイプのコレクションと、タイルが含む可能性のある全てのユーザ・インタフェースエレメントを実現する。タイルインスタンス、及び全ての含まれたUIエレメントはこのコンポーネント内で、タイルアセンブリとナビゲーションエンジンコンポーネントの指示に従って作成又は消去される。

【0062】・タイルアセンブリ及びナビゲーションエンジンコンポーネントは、総称的ユーザ・インタフェースモデルを実現する。アセンブリの作成中、このコンポーネントはタイル間関係を、UIエレメントとデータカーソルにより形成されたデータイベント伝播ネットワークを設定することにより、作成する。

【0063】・テーマハンドラコンポーネントは、タイルアセンブリ及びナビゲーションエンジンコンポーネントにより実現された総称的ユーザ・インタフェースモデルの、具体的ユーザ・インタフェースに対するマッピングを行う。テーマハンドラは、タイルアセンブリとナビゲーションエンジンの側と、ユーザ側との間のブリッジ役として機能する。定義されたテーマが必要とすれば、ハンドラは個々のタイル上で表示又は隠蔽(Show or Hide)動作や、グラフィカルでないユーザ・インタフェースの場合には同等の動作を行わせることができる。

【0064】・データイベントサービスコンポーネントは、個々のタイル、UIエレメント、及び同期サービスを1つにまとめる、言わば接着剤の役目を果たす。このコンポーネントにおけるデータカーソルオブジェクトは、オブジェクトのシーケンスと、シーケンス内の区別された位置の概念を実現する。区別された位置は“カーソル”と呼ばれ、その位置のオブジェクトは“カレントオブジェクト”と呼ばれる。データイベントサービスコンポーネント内のオブジェクトは、データ保存に関するイベントを、データカーソル及びUIエレメント間で伝播する。

【0065】・同期サービスコンポーネントは、モバイルクライアントデバイスと遠隔サーバ間でのデータ保存の同期を取る。このコンポーネントは、タイルハンドラコンポーネントから生じる、データ保存に対する変更を取り出し、記録し、最終的にサーバに送られるメッセージに組み入れる。他の方向では、同期サービスコンポーネントはサーバから、該サーバからの変更要求メッセージの形で変更を受け取り、該変更をデータ保存に適用する。受け取り動作は、データイベントを、変更に対応するデータイベントサービスコンポーネントに発行することにより完了する。

【0066】・データ保存サービスコンポーネントは、コンピューティングデバイスの継続的データ保存をカプセル化する。前記継続的保存は例えばリレーショナルデータベース管理システム、又は、ウィンドウズCEデータベース（マイクロソフト社、シアトル、WA）等のよりシンプルな設備である。

【0067】次の章では、より詳細な説明を行う。

【0068】6. 3. 2 アプリケーション文書サービスコンポーネント

アプリケーション文書サービスコンポーネントは、アプリケーション指定の符号化の理解のため、及び、その他のコンポーネントにおいて効率的に使用可能な形式で該指定を提示するために使用される。図8は、アプリケーション文書サービスコンポーネントによる、アプリケーションフラグメントの翻訳を図示したものである。同図において、ランタイムコンポーネントはタイル定義の指定を要求し、タイル定義のオブジェクト表示を返答として受け取る。総称的パーサ(Parser)はタイル定義フラグ

メントを読み取ると同時に、パーサイバントを、フラグメントのルートにあるエレメント、即ちタイルエレメントに固有のコンストラクタオブジェクトに発行する。このコンストラクタオブジェクトは、デバイス上のワーキングメモリ内のタイル定義のオブジェクト表示を作成し、このオブジェクト表示を、インタプリタのランタイムコンポーネントに送り返す。

【0069】好適な実施例では、フラグメントの内容に対する最初の参照の際にオブジェクト表示を要求し、該オブジェクトレファレンスを以後使用する時のためにキャッシュする。

【0070】6. 3. 3 インタプリタ変化型(variation)

アプリケーションシステムは、特別な環境又はシステム構成に特化されたインタプリタ変化型を提供できる。図2には2つのインタプリタ変化型が例示されている。

【0071】図2のクライアント側インタプリタは、完全にクライアント上で実行する。この好適な実施例は図7のアーキテクチャの完全な実現である。

【0072】図2のサーバ側インタプリタは、プレゼンテーション層を例外として、完全にサーバ上で実行する。この変化型は同期サービスコンポーネントを必要としない。このインタプリタ上では、WAPマイクロブラウザテーマは、デバイス上のネイティブ・マイクロブラウザを使用して、WML (Wireless Markup Language; WAP Consortium, www.wapforum.org) を実現する。ボイステーマは、音声認識及び音声合成ユーザ・インタフェースを実現する。この場合、プレゼンテーション層は電話上の音声入力及び出力である。

【0073】7 アプリケーション開発プロセス
ここでは、アプリケーションを作成するためにアプリケーション開発者が実行するプロセスを説明する。

【0074】7. 1 プロセスの概略

図1は、アプリケーション作成における各ステップを示す。アプリケーションは、アプリケーションシステム内のコンポーネントのコンテキスト及び使用(usage)により定義される。ワークベンチコンポーネントは、開発者が、アプリケーション指定言語、従ってアプリケーションモデルに適合したアプリケーション指定を作成することをサポートする。前記アプリケーションモデルは、特定の区画(Pane)、アセンブリ、及びナビゲーションに対応したユーザ・インタフェースモデルを含む。アプリケーション開発プロセスは、以下のステップを有する：アプリケーションシステムのワークベンチコンポーネントを使用して、アプリケーションの指定を含む文書を作成し；オプションとして、前記アプリケーション指定文書を特定のデバイス用に専用化し；アプリケーション指定文書を、インタプリタに適した形式に符号化する。

【0075】そして、アプリケーション指定は適切な記憶装置内に記憶され、要求に応じてインタプリタにより

検索及び実行される。

【0076】7. 2 プロセスの詳細

アプリケーション指定は、特定のアプリケーション指定言語に従ったXML文書である。そのような言語は、文書タイプ定義(DTD; Document Type Definition)又はXMLスキーマ等のメタ言語で指定される。ワークベンチは、アプリケーション開発者による前記文書の作成をサポートするXML文書である。このサポートのため、ワークベンチは、アプリケーション指定言語に適合した文書妥当性検査及びコンテキスト対応編集を行う。好適な実施例では、XMLスキーマメタ言語が使用される。

【0077】開発者は新規文書を作成する際、ワークベンチメニューから“新規”コマンドを呼び出す。その後、開発者は文書を変更するためには、XMLエレメントを付加ないし削除したり、それらエレメントの属性を付加、変更、又は削除する。エレメントの操作はツリービュー上のユーザ・インタフェース操作により行われ、他方、エレメント属性の操作は“エレメント属性パネル”を介して行われる。

【0078】文書内に新規インスタンスを挿入するためには、開発者はXMLツリービュー内のエレメントを選択し、ポップアップメニューを呼出す。そして開発者はメニューから、選択されたエレメントの前後いずれにおいて同じレベルでエレメントを付加するのか、又は選択されたエレメントの子として付加するのかを指定する。図4の例では、開発者はリストビューエレメントを選択し、子エレメントの挿入を指定した。ワークベンチはダイアログボックスにおいて、文書のこの場所の候補エレメントのリストを示す。候補エレメントは、アプリケーション指定言語の検査により自動的に決定される。

【0079】特定エレメントの属性の値を設定するためには、開発者はXMLツリービュー内のエレメントを選択する。ワークベンチは、アプリケーション指定言語を検査することにより、このエレメントに対する可能な全ての属性のリストを決定し、それらをエレメント属性パネル内に表示する。個々の属性のタイプ及びメタ属性に関する情報により、属性パネル上でのその視覚的外観が決定される。例えば、列挙型として定義された属性は、図6に示すように、列挙された数値の集合を含むドロップダウンリストとして表示できる。

【0080】アプリケーション開発者はこのようにして、アプリケーション指定文書を作成する。このプロセスの好適な実施例の1つはプロトタイプ化と呼ばれ、エンドユーザの理解及びデザインに関するフィードバックを得るため、まずユーザ・インタフェース部分が作成される。データモデル部分はその後で作成される。別の好適な実施例では、データモデルがアプライオリに分かっている場合、文書のその部分が最初に作成され、続いてユーザ・インタフェース部分が作成される。

【0081】プロトタイプ化の実施例は、アプリケーシ

ョンの提示をインタプリタ上で近似できるプレビューアコンポーネントにより可能となる。プレビューアコンポーネントはこのシミュレーションを、不完全で従ってインタプリタ自体では実行不可能な文書に対して行うことができる。個々のプレビューアは特定レイアウトテーマのシミュレーションを実現するので、また、複数のプレビューアは同時に表示可能であるので、開発者は、アプリケーションのそれぞれの目標デバイス上での提示を同時に確認することができる。ワークベンチのアーキテクチャはまた、アプリケーション文書への変更はいずれも全てのコンポーネントに同報通信されるよう構成されているので、個々の目標デバイスにおける提示の確認を、アプリケーションの作成又は変更中に行うことができる。

【0082】アプリケーション指定文書の作成が終了した後、開発者はワークベンチの妥当性検査機能を使用して、インタプリタが正確に文書を実行できることを確認

する。妥当性検査機能は言語定義によってもドライブされる文書は最後に変換・符号化され（図1参照）、インタプリタにより使用可能なように記憶される。

【0083】8 モバイルデータアプリケーションフレームワーク（MDF）アプリケーションシステム
この章では、アプリケーションシステムの好適な実施例、すなわちモバイルデータアプリケーションフレームワーク（MDF）を説明する。

【0084】8. 1 MDFアプリケーション指定言語
第5. 2章で説明したユーザ・インタフェースモデルに対応する言語が、表1に示されている。

【0085】表1に示したMDFアプリケーション指定言語はXML対応言語であり、XMLスキーマ（現時点ではマイクロソフト社による、ドラフトW3C標準の具体化に基づく）により定義される。

【0086】

【表1】

MDFアプリケーション指定言語はXML対応言語であり、XMLスキーマ（現時点ではドラフト W3C標準のマイクロソフト製品を利用した実現に基づく）により定義される。

<!-- XML Schema Extensibility for MDF schemas -->

```
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes">
```

<!-- Attribute to support subtyping of document idrefs.
Usage in schema:

```
<AttributeType name="idref" ext:refType="xxx"
  ext:refScope="yyy">
  where
    xxx is/are the name(s) of tag(s) to which the
    attribute can refer, and
    yyy is/are the name(s) of the tag(s) which are the
    scope for the reference.
    Ids will only be matched if lie outside the
    innermost containing element with nodename from the set
    of tags yyy.
    The document processing application is responsible
    for enforcing the subtyping.
```

```
<AttributeType name="refType" dt:type="string"/>
<AttributeType name="refScope" dt:type="string"/>
```

<!-- Support layout

displayUnit: attributed to containers that can be shown or hidden

mergeable = could be combined with parent
Elements
unmergeable = never combined with parent
elements

```
<AttributeType name="displayUnit" dt:type="enumeration"
  dt:values="unmergeable mergeable"/>
```

<!-- Support for fragmentation -->

```
<AttributeType name="fragment" dt:type="boolean"/>
```

<!-- Support for translation -->

```
<AttributeType name="translatable" dt:type="boolean"/>
```

<!-- Descriptive text -->

```
<AttributeType name="description" dt:type="string"/>
```

<!-- Support for allowing identification of layout-related
attributes and tags -->

```
<AttributeType name="layout" dt:type="enumeration"
  dt:values="all VGA HVGA PPC"/>
```

<!--

Temporary extension-attributes, while waiting for
Microsoft to upgrade their XML Schema support!!

```
<AttributeType name="length" dt:type="int"/>
<AttributeType name="minLength" dt:type="int"/>
<AttributeType name="maxLength" dt:type="int"/>
<!-- Next extension-attribute, if present, overrides dt:type -->
<AttributeType name="type" dt:type="enumeration"
  dt:values="bin.hex bin.base64 filename icon bitmap"/>
```

<!-- Elements to support data-binding.

Usage in schema:

```
<Element type="aaa" ...><SupportsType
  fieldType="xxx"/></SupportsType ...>
  Indicates that 'aaa' can be bound to fields with type
  'xxx', etc.
```

```
<AttributeType name="dataBinding" dt:type="enumeration"
  dt:values="table field"/>
<AttributeType name="control" dt:type="boolean"/>
```

```
<AttributeType name="fieldType" dt:type="enumeration"
  dt:values="string number integer float boolean date time
  dateTime enumeration"/>
<Element type="SupportsType">
  <Attribute type="fieldType" required="yes"/>
</Element>
</Schema>
```

<!-- MDF Application Schema, \$Headers -->

```
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes"
  xmlns:ext="x-schema:SchemaExtension.xsd">
```

```
<!--_mdf_top element=GUI version=1.07-->
```

<Element type="idMap" ext:description="Placeholder
for the WBXML id-Offset map">

```
<AttributeType name="pointers" dt:type="string"/>
<Attribute type="pointers" required="yes"/>
</Element>
```

```
<Element type="inlineCode" dt:type="string"
  content="textOnly"/>
```

```
<AttributeType name="constraintType"
  dt:type="enumeration" dt:values="validity too bar"/>
<Element type="Constraint" content="textOnly">
  <Attribute type="constraintType" required="yes"/>
  <element type="inlineCode" minOccurs="1"
  maxOccurs="1"/>
</Element>
```

```
<AttributeType name="functionName" dt:type="string"/>
<AttributeType name="event" dt:type="string"/>
<AttributeType name="argName" dt:type="string"/>
<Element type="Argument">
  <Attribute type="argName" required="yes"/>
  <Attribute type="argDefault" required="no"/>
</Element>
```

```
<Element type="Function">
  <Attribute type="functionName" required="yes"/>
  <element type="Argument" minOccurs="1"/>
</Element>
```

```
<Element type="FunctionCell">
  <Attribute type="functionName" required="yes"/>
  <element type="Argument" minOccurs="1"/>
</Element>
```

```
<AttributeType name="idTable" dt:type="id"/>
<AttributeType name="refTable" dt:type="idref"
  ext:refType="Table" ext:databinding="table"/>
<AttributeType name="idField" dt:type="id"/>
<AttributeType name="dsn" dt:type="string"/>
<AttributeType name="refField" dt:type="idref"
  ext:refType="Field"/>
<AttributeType name="refFields" dt:type="idrefs"
  ext:refType="Field"/>
<AttributeType name="type" dt:type="enumeration"
  dt:values="string integer dateTime date time fixedDecimal
  boolean"/>
<AttributeType name="length" dt:type="int"/>
<AttributeType name="decimalLength" dt:type="int"/>
<AttributeType name="defaultLabel" dt:type="string"/>
<AttributeType name="constant" dt:type="string"/>
<AttributeType name="paramIndex" dt:type="int"/>
```

```
<AttributeType name="id" dt:type="id"/>
<AttributeType name="title" dt:type="string"
  ext:translatable="1"/>
```

```
<AttributeType name="importance" dt:type="enumeration"
  dt:values="mandatory high normal low"/>
```

<!-- Bitmap and icon support -->

【0087】

【表2】

```

<AttributeType name='iconFile' dt:type='string'
ext:type='filename'/>
<AttributeType name='blobFile' dt:type='string'
ext:type='filename'/>

<!-- new: Sorting -->
<!-- Disabled for V1.0
<AttributeType name='order' dt:type='enumeration'
dt:values='ascending descending'/>
<ElementType name='SortKey' content='empty'>
  <attribute type='refField' required='yes'/>
  <attribute type='order' required='yes'/>
</ElementType>

<ElementType name='SortSpec' content='allOnly'>
  <attribute type='id' required='yes'/>
  <attribute type='title' required='yes'/>
  <element type='SortKey' minOccurs='1' maxOccurs='1'/>
</ElementType>

<!-- new: Hyperlinking.
We might also specify hyperlinks by more complex
queries, or by Constraints
-->
<AttributeType name='ownField' dt:type='idref'
ext:refType='Field' ext:databinding='field'/>
<AttributeType name='desField' dt:type='idref'
ext:refType='Field' ext:databinding='field'/>
<AttributeType name='desTitleSet' dt:type='idref'
ext:refType='TitleSet'/>
<ElementType name='HyperLink' content='allOnly'>
  <attribute type='ownField' required='yes'/>
  <attribute type='desField' required='yes'/>
  <attribute type='desTitleSet' required='yes'/>
  <element type='VoiceCaption' minOccurs='0'
maxOccurs='1'/>
</ElementType>

<AttributeType name='displayField' dt:type='idref'
ext:refType='Field' ext:databinding='field'/>
<ElementType name='Indirect' content='empty'>
  <attribute type='refTable' required='yes'/>
  <attribute type='refField' required='yes'/>
  <attribute type='displayField' required='yes'/>
</ElementType>

<!-- START DataModel schema -->
<ElementType name='ContentSelector' content='empty'>
  <attribute type='refField' required='yes'/>
  <attribute type='constant' required='yes'/>
</ElementType>

<ElementType name='ParameterSelector' content='empty'>
  <attribute type='refField' required='yes'/>
  <attribute type='parameter' required='yes'/>
</ElementType>

<ElementType name='Checktable' content='allOnly'>
  <attribute type='refTable' required='yes'/>
  <attribute type='refField' required='yes'/>
  <group order='many'>
    <element type='ConstantSelector' minOccurs='0'/>
    <element type='ParameterSelector' minOccurs='0'/>
  </group>
</ElementType>

<!-- Model the relationships directly -->
<ElementType name='Relation' content='allOnly'>
  <AttributeType name='table1' dt:type='idref'
ext:refType='Table'/>
  <AttributeType name='table2' dt:type='idref'
ext:refType='Table'/>
  <AttributeType name='field1' dt:type='idref'
ext:refType='Field'/>
  <AttributeType name='field2' dt:type='idref'
ext:refType='Field'/>
  <AttributeType name='cardinality1' dt:type='enumeration'
dt:values='1 n'/>
  <AttributeType name='cardinality2' dt:type='enumeration'
dt:values='1 n'/>

```

【0088】

```

<AttributeType name='dependentObjects'
dt:type='enumeration' dt:values='table1 table2 neither'
default='neither'/>
  <attribute type='id' required='yes'/>
  <attribute type='table1' required='yes'/>
  <attribute type='table2' required='yes'/>
  <attribute type='field1' required='yes'/>
  <attribute type='field2' required='yes'/>
  <attribute type='cardinality1' required='yes'/>
  <attribute type='cardinality2' required='yes'/>
  <attribute type='dependentObjects' required='no'/>
</AttributeType>
<AttributeType name='parentRelation' dt:type='idref'
ext:refType='Relation MainTitle SubTitle'
ext:refScope='ThisSet'/>

<!-- Linking: Another possibility is to use constraints.
A SubTitle's ownField and parentField are replaced by an
attribute link which references the constraint.
The FK constraint is used in both directions, depending on
context.
-->
<AttributeType name='baseTable' dt:type='idref'
ext:refType='Table'/>
<AttributeType name='baseField' dt:type='idref'
ext:refType='Field'/>
<AttributeType name='referringTable' dt:type='idref'
ext:refType='Table'/>
<AttributeType name='referringField' dt:type='idref'
ext:refType='Field'/>
<ElementType name='Constraint_ForeignKey'
content='allOnly'>
  <attribute type='id' required='yes'/>
  <attribute type='baseTable' required='yes'/>
  <attribute type='baseField' required='yes'/>
  <attribute type='referringTable' required='yes'/>
  <attribute type='referringField' required='yes'/>
</ElementType>

<AttributeType name='link' dt:type='idref'
ext:refType='Constraint_ForeignKey'/>
-->

<AttributeType name='value' dt:type='string'/>
<ElementType name='ValueListElement' content='empty'>
  <attribute type='value' required='yes'/>
</ElementType>

<ElementType name='ValueList' content='allOnly'>
  <element type='ValueListElement' minOccurs='1'
maxOccurs='1'/>
</ElementType>

<ElementType name='ValueRange' content='empty'>
  <AttributeType name='lowerBound' dt:type='string'/>
  <AttributeType name='lowerBoundInclusive'
dt:type='boolean' default='1'/>
  <AttributeType name='upperBound' dt:type='string'/>
  <AttributeType name='upperBoundInclusive'
dt:type='boolean' default='1'/>
  <attribute type='lowerBound' required='no'
ext:description='Not present means not bound'/>
  <attribute type='lowerBoundInclusive' required='no'
ext:description='Not present means not bound'/>
  <attribute type='upperBound' required='no'
ext:description='Not present means not bound'/>
  <attribute type='upperBoundInclusive' required='no'
ext:description='Not present means not bound'/>
</ElementType>

<ElementType name='DomainCheck' order='any'
content='allOnly'>
  <element type='Checktable' minOccurs='0'
maxOccurs='1'/>
  <element type='ValueList' minOccurs='0' maxOccurs='1'/>
  <element type='ValueRange' minOccurs='0'
maxOccurs='1'/>
</ElementType>

<ElementType name='Field' content='allOnly'>
  <attribute type='idField' required='yes'/>
  <attribute type='type' required='yes'/>
  <attribute type='length' required='no'/>
  <attribute type='decimalLength' required='no'/>
  <attribute type='defaultLabel' required='no'/>
  <AttributeType name='nullAllowed' dt:type='boolean'
dt:type='enumeration' dt:values='no integer GUID'
dt:values='no integer GUID' default='0'/>
  <attribute type='nullAllowed' required='no' default='0'/>
  <attribute type='autoGenerate' required='no' default='no'/>

```

【表3】

```

<element type="DomainCheck" minOccurs="0"
maxOccurs="1"/>
</ElementTypes>

<ElementTypes name="Index" content="empty">
<attribute type="refField" required="yes"/>
</ElementTypes>

<AttributeType name="externalKey" dtype="idref"
extrefType="Field" extrefScope="Table"/>
<ElementTypes name="Table" content="tableOnly">
<attribute type="idTable" required="yes"/>
<attribute type="idnt" required="yes"/>
<attribute type="externalKey" required="no"/>
<element type="Field" minOccurs="1" maxOccurs="1"/>
<element type="Index" minOccurs="0"/>
</ElementTypes>

<ElementTypes name="DataModel" content="tableOnly"
extrefType="Table">
<group order="many" minOccurs="1">
<element type="Table"/>
<element type="Relation"/>
</group>
</ElementTypes>
<!-- END DataModel schema -->

<!--
Currently supported:
- Checkables, in the data model

To do:
- Complete modelling of controls
- Add support for popup frames

- Add Workflow support (see other design documentation):
Rules of form: if CNF(primitive constraints) then
useOf(Field,Value) > SetOf(Variable,Value)
Primitive constraint: (field,operator,value/field)

- Tradeoff between VBScript and built-in functionality is:
- built-in is descriptive -> can be understood and
manipulated
- complexity of program vs general VBScript approach
- simplicity of customization vs. discontinuity

To Review

- Support for scripting
=> keep simple, to ease implementation, and also
validation (but reduces error-checking!)
- return value has fixed name in calls
- script won't support types, so pointless to indicate in
grammar
- ? use Perl-like calling scheme, where formal
parameters have fixed names.
=> need to work out requirements a bit better

Notes on IE5 support:
- ?enumeration type not supported
- mode="open" is default
-->

<!--[CDATA[
One way of handling Complex fields:
<Complex label="Address">
<Group compress="yes" separator=" " >
<ComplexSub datafield="Addr1"/>
<Break/>
<ComplexSub datafield="Addr2"/>
<Break/>
<Group compress="yes" separator=" " >
<ComplexSub datafield="City"/>
<ComplexSub datafield="Region"/>
</Group>
</Complex>
]]>

<AttributeType name="idTitle" dtype="id"/>
<AttributeType name="text" dtype="string"/>
<AttributeType name="refTitle" dtype="idref"
extrefType="table"/>

```

```

<AttributeType name="datafield" dtype="idref"
extrefType="Field" extrefScope="table"/>
<AttributeType name="parentField" dtype="idref"
extrefType="Field"/>
<AttributeType name="sortBy" dtype="idref"
extrefType="SortSpec"/>
<AttributeType name="multiline" dtype="boolean"/>
<AttributeType name="navigationBar" dtype="boolean"/>

<AttributeType name="label" dtype="string"/>
<AttributeType name="visible" dtype="boolean"/>
<AttributeType name="navigate" dtype="boolean"/>
<AttributeType name="readOnly" dtype="boolean"/>
<AttributeType name="mandatory" dtype="boolean"/>

<!-- For element Row -->
<AttributeType name="row" dtype="idref" extrefType="table"
extrefScope="table"/>
<AttributeType name="colspan" dtype="int"
extrefType="table"/>
<AttributeType name="align" dtype="enumeration"
dtvalues="left center right" extrefType="table"/>
<AttributeType name="hspace" dtype="int"
extrefType="table"/>

<!-- For element GridPage -->
<AttributeType name="wrap" dtype="boolean"
extrefType="table"/>
<AttributeType name="cols" dtype="int" extrefType="table"
extrefScope="table"/>

<ElementTypes name="LayoutAttr" content="tableOnly"
extrefType="table">
<attribute type="colspan"/>
<attribute type="align"/>
<attribute type="hspace"/>
</ElementTypes>

<ElementTypes name="GridPage" content="tableOnly"
extrefType="table">
<attribute type="cols"/>
<attribute type="wrap" required="yes" default="1"/>
<element type="Row" minOccurs="0" maxOccurs="1"/>
</ElementTypes>

<!-- Code -->
<ElementTypes name="EventHandler" content="tableOnly">
<attribute type="event"/>
<group order="one">
<element type="InitCode"/>
<element type="FunctionCall"/>
</group>
</ElementTypes>

<AttributeType name="format" dtype="string"
extrefType="table"/>
<AttributeType name="fieldList" dtype="idref"
extrefType="table"/>
<ElementTypes name="Caption">
<attribute type="format" required="yes"/>
<attribute type="fieldList" required="no"/>
</ElementTypes>
<ElementTypes name="VoiceCaption">
<attribute type="format" required="yes"/>
<attribute type="fieldList" required="yes"/>
</ElementTypes>

<!-- Tiberts, Tiles, etc. -->
<AttributeType name="targetFrame" dtype="enumeration"
dtvalues="1 2 12 3 4 34 1234"/>
<AttributeType name="displayGroup" dtype="int"/>
<ElementTypes name="SubTitle">
<attribute type="title" required="yes"/>
<attribute type="id" required="no"/>
<attribute type="refTitle" required="yes"/>
<attribute type="targetFrame" required="yes"/>
<attribute type="displayGroup" required="no"/>
<attribute type="parentRelation" required="yes"/>
<attribute type="sortBy" required="no"/>
<attribute type="visible" required="no" default="1"/>
<attribute type="importance" required="no"
default="normal"/>
<element type="Caption" minOccurs="0"
maxOccurs="1"/>
<element type="VoiceCaption" minOccurs="0"
maxOccurs="1"/>
<element type="SubTitle"/>

```

[0089]

[表4]

```

</ElementTypes>
<ElementType name="MainTitle">
  <attribute type="title" required="yes"/>
  <attribute type="id" required="no"/>
  <attribute type="refId" required="yes"/>
  <attribute type="targetFrame" required="yes"/>
  <attribute type="displayGroup" required="no"/>
  <attribute type="body" required="no"/>
  <element type="Caption" minOccurs="0" maxOccurs="1"
    ext:description="Overrides the attribute"/>
  <element type="VoiceCaption" minOccurs="0"
    maxOccurs="1" ext:description="Overrides the attribute"/>
  <element type="SubTitle" maxOccurs="1"/>
</ElementType>
<ElementType name="DataControl">
  <attribute type="datafield" required="yes"/>
</ElementType>
<AttributeType name="displayCharacters" datatype="int"/>
<ElementType name="Column">
  <attribute type="title" required="yes"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <attribute type="displayCharacters" required="no"/>
  <element type="Indirect" minOccurs="0" maxOccurs="1"/>
  <element type="VoiceCaption"/>
</ElementType>
<!-- Controls -->
<!-- Next are advisory; may be overridden by Layout or
  Engine components -->
<AttributeType name="minDisplayCharacters"
  datatype="int"/>
<ElementType name="Label" content="empty"
  ext:control="1">
  <!-- #Extends(DataControl) -->
  <attribute type="text" required="yes"/>
</ElementType>
<ElementType name="TextBox" content="allOnly"
  ext:control="1">
  <!-- #Extends(DataControl) -->
  <attribute type="label" required="no"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="minDisplayCharacters" required="no"/>
  <!-- #attribute type="type" required="no" default="string"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <attribute type="readonly" required="no" default="0"/>
  <attribute type="mandatory" required="no" default="0"/>
  <attribute type="multiline" required="no" default="0"/>
  <element type="HyperLink" minOccurs="0"
    maxOccurs="1"/>
  <element type="Indirect" minOccurs="0" maxOccurs="1"/>
  <element type="LayoutAttr" minOccurs="0"
    maxOccurs="1"/>
  <element type="Event-handler" minOccurs="0"/>
  <ext:SupportsType fieldType="string"/>
  <ext:SupportsType fieldType="integer"/>
</ElementType>
<ElementType name="ComboBox" content="allOnly"
  ext:control="1">
  <!-- #Extends(DataControl) -->
  <attribute type="label" required="no"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="minDisplayCharacters" required="no"/>
  <AttributeType name="showDescription"
    datatype="boolean"/>
  <attribute type="showDescription" required="no"
    default="0"/>
  <!-- #attribute type="type" required="no" default="string"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <element type="HyperLink" minOccurs="0"
    maxOccurs="1"/>
  <element type="Indirect" minOccurs="0" maxOccurs="1"/>
  <element type="LayoutAttr" minOccurs="0"
    maxOccurs="1"/>

```

```

  <element type="Event-handler" minOccurs="0"/>
  <ext:SupportsType fieldType="string"/>
  <ext:SupportsType fieldType="integer"/>
</ElementType>
<!-- OptionGroups and Checkboxes -->
<ElementType name="Option" content="allOnly"
  ext:control="1">
  <attribute type="text" required="yes"/>
  <attribute type="value" required="yes"/>
  <attribute type="refId" required="no" ext:description="The
    title to show Page on. None means the parent of the Option
    control"/>
  <element type="Page" minOccurs="0" maxOccurs="1"
    ext:description="Page to show on this option being
    selected"/>
</ElementType>
<AttributeType name="default" datatype="string"/>
<ElementType name="OptionGroup" ext:control="1">
  <attribute type="title" required="no"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <attribute type="readonly" required="no" default="0"/>
  <attribute type="minDisplayCharacters" required="no"/>
  <attribute type="default" required="no"/>
  <element type="Option" minOccurs="1" maxOccurs="1"/>
  <ext:SupportsType fieldType="string"/>
  <ext:SupportsType fieldType="integer"/>
</ElementType>
<ElementType name="OnTrue">
  <attribute type="refId" required="yes"/>
  <attribute type="Visible" required="no" default="0"/>
  <attribute type="navigate" required="no"/>
</ElementType>
<ElementType name="OnFalse">
  <attribute type="refId" required="yes"/>
  <attribute type="Visible" required="no" default="0"/>
  <attribute type="navigate" required="no"/>
</ElementType>
<ElementType name="CheckBox" content="allOnly"
  ext:control="1">
  <!-- #Extends(DataControl) -->
  <attribute type="text" required="yes"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <attribute type="readonly" required="no" default="0"/>
  <attribute type="minDisplayCharacters" required="no"/>
  <element type="OnFalse" minOccurs="0" maxOccurs="1"/>
  <element type="OnTrue" minOccurs="0" maxOccurs="1"/>
  <element type="HyperLink" minOccurs="0"
    maxOccurs="1"/>
  <element type="LayoutAttr" minOccurs="0"
    maxOccurs="1"/>
  <element type="Event-handler" minOccurs="0"/>
  <ext:SupportsType fieldType="boolean"/>
</ElementType>
<ElementType name="Monitor" content="allOnly"
  ext:control="1">
  <attribute type="label" required="no"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <attribute type="readonly" required="no" default="0"/>
  <attribute type="minDisplayCharacters" required="no"/>
  <attribute type="mandatory" required="no" default="0"/>
  <!-- #Extends(DataControl) -->
  <element type="Indirect" minOccurs="0" maxOccurs="1"/>
  <element type="LayoutAttr" minOccurs="0"
    maxOccurs="1"/>
  <element type="Event-handler" minOccurs="0"/>
  <ext:SupportsType fieldType="string"/>
</ElementType>
<ElementType name="DateTime" content="allOnly"
  ext:control="1">
  <!-- #Extends(DataControl) -->
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>

```

```

<attribute type='importance' required='no'
default='normal'/>
<attribute type='readonly' required='no' default='0'/>
<attribute type='mandatory' required='no' default='0'/>
<attribute type='label' required='no'/>
<attribute type='minDisplayCharacters' required='no'/>
<element type='HyperLink' minOccurs='0'
maxOccurs='1'/>
<element type='LayoutAttr' minOccurs='0'
maxOccurs='1'/>
<ext:SupportType fieldType='date'/>
<ext:SupportType fieldType='time'/>
<ext:SupportType fieldType='datetime'/>
</Element>

<!-- Special controls: Address -->
<AttributeType name='addr1' dtype='string'/>
<AttributeType name='addr2' dtype='string'/>
<AttributeType name='city' dtype='string'/>
<AttributeType name='region' dtype='string'/>
<AttributeType name='country' dtype='string'/>
<AttributeType name='postal' dtype='string'/>

<ElementType name='Address' content='eltOnly'
ext:control='1'/>
<!-- # Extends (DateControl) -->
<attribute type='id' required='yes'/>
<attribute type='importance' required='no'
default='normal'/>
<attribute type='readonly' required='no' default='0'/>
<attribute type='mandatory' required='no' default='0'/>
<attribute type='minDisplayCharacters' required='no'/>
<attribute type='label' required='no'/>

<attribute type='addr1' required='no'/>
<attribute type='addr2' required='no'/>
<attribute type='city' required='no'/>
<attribute type='region' required='no'/>
<attribute type='country' required='no'/>
<attribute type='postal' required='no'/>

<element type='EventHandler' minOccurs='0'/>
</Element>

<!-- should MonthlyCalendar & DaySchedule be controls
or Tiles? -->
<ElementType name='MonthlyCalendar' content='eltOnly'>
<attribute type='id' required='yes'/>
<attribute type='datafield' required='yes'/>
<ext:SupportType fieldType='date'/>
<ext:SupportType fieldType='datetime'/>
</Element>

<ElementType name='DaySchedule' content='eltOnly'>
<attribute type='id' required='yes'/>
<attribute type='datafield' required='yes'/>
<ext:SupportType fieldType='date'/>
<ext:SupportType fieldType='datetime'/>
</Element>

<AttributeType name='displayFormat' dtype='string'/>
<AttributeType name='ComplexField'>
<attribute type='id' required='yes'/>
<attribute type='datafield' required='yes'/>
<attribute type='type' required='no' default='string'/>
<attribute type='importance' required='no'
default='normal'/>
<attribute type='readonly' required='no' default='0'/>
<attribute type='mandatory' required='no' default='0'/>
<attribute type='minDisplayCharacters' required='no'/>

<attribute type='displayFormat' required='yes'/>
<group minOccurs='1' maxOccurs='*' order='many'>
<!-- element type=DataControl -->
<element type='Address'/>
<element type='DateTime'/>
<element type='CheckBox'/>
<element type='ComboBox'/>
<element type='ComplexField'/>
<element type='Memo'/>
<element type='OptionGroup'/>
<element type='TextBox'/>
</group>
</Element>

```

```

<!-- Commands -->
<AttributeType name='command' dtype='enumeration'
dt:values='faxdoc telephone'/>
<ElementType name='StandardCommand'>
<attribute type='command' required='yes'/>
<element type='Argument' minOccurs='1'/>
</Element>
<ElementType name='AddInCommand'>
<attribute type='functionName' required='yes'/>
<element type='Argument' minOccurs='1'/>
</Element>

<AttributeType name='topMenu' dtype='string'/>
<ElementType name='MenuItem' content='eltOnly'>
<attribute type='label' required='yes'/>
<attribute type='topMenu' required='yes'/>
<group minOccurs='1' order='one'>
<element type='StandardCommand'/>
<element type='AddInCommand'/>
</group>
</Element>

<!-- Shareable elements -->
<ElementType name='Row' content='eltOnly'
ext:displayUnit='mergable' ext:layout='elt'>
<group minOccurs='0' maxOccurs='*' order='many'>
<!-- element type=DataControl -->
<element type='Address'/>
<element type='DateTime'/>
<element type='CheckBox'/>
<element type='ComboBox'/>
<element type='ComplexField'/>
<element type='Memo'/>
<element type='OptionGroup'/>
<element type='TextBox'/>
</group>
</Element>

<ElementType name='Page' content='eltOnly'
ext:displayUnit='mergable' ext:layout='elt'>
<attribute type='id'/>
<attribute type='title' required='yes'
ext:description='Used if the Page is not merged'/>
<group minOccurs='0' maxOccurs='*' order='many'>
<!-- element type=DataControl -->
<element type='Address'/>
<element type='DateTime'/>
<element type='CheckBox'/>
<element type='ComboBox'/>
<element type='ComplexField'/>
<element type='Memo'/>
<element type='OptionGroup'/>
<element type='TextBox'/>
</group>
</Element>

<ElementType name='Tile' content='eltOnly'
ext:displayUnit='unmergable' ext:fragment='1'>
<attribute type='idTile' required='yes'/>
<attribute type='refTable' required='yes'/>
</Element>

<ElementType name='FrameTile' content='eltOnly'
ext:displayUnit='unmergable' ext:fragment='1'>
<attribute type='idTile' required='yes'/>
<attribute type='refTable' required='yes'/>
<!-- # Extends (Tile) -->
<group minOccurs='1' order='many'>
<element type='Page'/>
<element type='GridPage'/>
<element type='EventHandler'/>
</group>
</Element>

<AttributeType name='defaultSearchField' dtype='idref'
ext:refType='Field'/>
<ElementType name='ListViewTile' content='eltOnly'
ext:displayUnit='unmergable' ext:fragment='1'>
<attribute type='idTile' required='yes'/>
<attribute type='refTable' required='yes'/>
<!-- # attribute type=defaultSearchField required='no' -->
<!--
<element type='Search' minOccurs='0'
maxOccurs='1'/>
<element type='HyperLink' minOccurs='0'
maxOccurs='1'/>
<element type='ValueCaption' minOccurs='0'
maxOccurs='1' ext:description='Describes the row'/>

```

[0091]

[表6]


```

<!-- element type=SortSpec minOccurs=0
maxOccurs=1 -->
<!-- Extends(Tile) -->
<group minOccurs="1" maxOccurs="1">
  <element type="Column"/>
</group>
</ElementTypes>

<!-- these two Field setting legs assume its the
Main Tile table we are manipulating -->
<!-- Copy from (current table?) to field in new object -->
<Element type="CopyField" content="allOnly">
  <attribute type="destField" required="yes"/>
  <attribute type="srcField" required="yes"/>
</ElementTypes>

<!-- Set field in new object -->
<Element type="FieldValue" content="allOnly">
  <attribute type="destField" required="yes"/>
  <attribute type="value" required="yes"/>
</ElementTypes>

<Attribute type="refModify" datatype="idref"
extensible="false"/>
<Element type="OnNew" content="allOnly">
  <attribute type="destTileSet" required="yes"/>
  <element type="CopyField" minOccurs="0"
maxOccurs="1"/>
  <element type="FieldValue" minOccurs="0"
maxOccurs="1"/>
</ElementTypes>

<Element type="OnCopy" content="allOnly">
  <attribute type="destTileSet" required="yes"/>
  <element type="CopyField" minOccurs="0"
maxOccurs="1"/>
  <element type="FieldValue" minOccurs="0"
maxOccurs="1"/>
</ElementTypes>

<Element type="TileSet" content="allOnly"
extensible="false">
  <attribute type="title" required="yes"/>
  <attribute type="icon" required="yes"/>
  <attribute type="refModify" required="no"/>
  <attribute type="navigationBar" required="no"
default="0"/>
  <attribute type="readOnly" required="no" default="1"/>
  <attribute type="bitmapFile" required="no"/>
  <element type="OnNew" minOccurs="0"
maxOccurs="1"/>
  <element type="OnCopy" minOccurs="0"
maxOccurs="1"/>
  <element type="MainTile" minOccurs="1"
maxOccurs="1"/>
</ElementTypes>

<Element type="TileSet" content="allOnly">
  <group minOccurs="1">
    <element type="TileSet">
</group>
</ElementTypes>

<!-- Shareable element pools -->
<Element type="Tables" content="allOnly">
  <element type="Table" minOccurs="1"/>
</ElementTypes>

<Element type="Tiles" content="allOnly">
  <group minOccurs="1" order="many">
    <element type="ListViewTile"/>
    <element type="FrameTile"/>
  </group>
</ElementTypes>

<Element type="Functions" content="allOnly">
  <group minOccurs="0">
    <element type="Function"/>
  </group>
</ElementTypes>

<!-- Middleware Information -->
<Attribute type="scenarioName" datatype="string"
ext:maxLength="3" ext:minLength="10"/>
<Attribute type="scenarioId" datatype="string"
ext:length="5"/>
<Element type="ScenarioSegment">
  <attribute type="structId" required="yes"/>
  <attribute type="refTable" required="yes"/>
</ElementTypes>

```

```

<Element type="Scenario">
  <attribute type="scenarioName" required="yes"/>
  <attribute type="id" required="no"/>
  <element type="ScenarioSegment" minOccurs="1"
maxOccurs="1"/>
</ElementTypes>

<Element type="Middleware">
  <element type="Scenario" minOccurs="1"
maxOccurs="1"/>
</ElementTypes>

<!-- Top level -->
<Element type="GUI" content="allOnly">
  <attribute type="title" required="yes"/>
  <attribute type="iconFile" required="no"/>
  <element type="DataModel" minOccurs="1"
maxOccurs="1"/>
  <element type="Middleware" minOccurs="0"
maxOccurs="1"/>
  <element type="Tiles" minOccurs="1" maxOccurs="1"/>
  <element type="Functions" minOccurs="0"
maxOccurs="1"/>
  <element type="TileSets" minOccurs="0"
maxOccurs="1"/>
</ElementTypes>
</Schema>

```

【0092】8.2 MDFワークベンチ

MDF用ワークベンチは、開発者に対して、MDF言語定義（XMLスキーマ）からドライブされたコンテキスト対応ガイダンスを提供する。図4の中央に、開発者の「ノード挿入」コマンドに回答して、候補エレメントのリストを表示するポップアップウィンドウが示されている。候補リストは、ワークベンチによりMDF言語定義を検査することにより決定される。

【0093】図4の下部に、アプリケーションが、サポートされているレイアウトテーマの1つにおいてどのように見えるかが分かるプレビューが示されている。ここでは、レイアウトテーマはハンドヘルドPC（Handheld PC）である。ワークベンチは、テーマの外観と動作に近似したスクリーンを提供することにより、アプリケーシ

ョン開発者がアプリケーションが実装された時の視覚的イメージを得ることができるようにする。

【0094】8.3 MDFインタプリタ

総称的インタプリタは第6.3章に説明してある。

【0095】アプリケーションの第5ステップ、総称的UIモデルの実現

第6.1章で説明したシステム作成プロセスは、C++等のオブジェクトプログラミング言語を使用して実行可能である。同様に、第6ステップである、MDFテーマの実現も、このプログラミング言語を使用して実行できる。

【0096】8.3.1 MDFレイアウトテーマ

この章では、UIモデルが、特定デバイスのスクリーンレイアウト（「テーマ」）に左右されないことを示す。

1つのMDFアプリケーション定義が、3つのGUIテーマと共に示されている。これら3つのテーマは、ハーフVGA(Half-VGA)ハンドヘルドPC、パーム(Palm)PC、及びVGAハンドヘルドPCであり、対応するスクリーンサイズを有するウィンドウズCE(マイクロソフト社、シアトル、WA)に適用可能である。

【0097】1.1 ハンドヘルドPC(ハーフVGAサイズ)

図9に、ハーフVGAハンドヘルドPC(縦240ピクセル、横640ピクセル)スクリーンレイアウトが示されている。総称的UIモデルのテーマに対するマッピングは以下のように行われる:

(a) 最上位タイルセットアセンブリのタイトルが、左上のナビゲーションバーの形で表示される;

(b) ナビゲーションスタックがリストとして左下に表示される。エントリをクリックすることにより、選択されたエントリより奥にある全てのエントリが除外される。さらに、逆向きの矢印によりスタックから最上位の要素を除外することができる。

【0098】(c) リストビュータイル及びフレームタイルテキストボックス要素は関連したハイパーリンクを有する場合がある。ハイパーリンクをクリックすることにより該ハイパーリンクが呼び出され、新しいタイルセットが作成され、ナビゲーションスタックの最上位に付加される。今までのカレントタイルセットは隠され、新規タイルセットが表示される。

【0099】(d) カレントタイルセットのメインタイルは即座に中央ウィンドウのナビゲーションバーの右側に示され、サブタイルが右側のタブストリップ内に示される。

【0100】1.2 パームPC

パームPC(縦320ピクセル、横240ピクセル)スクリーンレイアウトは、MDFプロトタイプによりサポートされる最小のGUIレイアウトである。図10には、外観が示されている。ハンドヘルドPCテーマとは対照的に、最上位タイルセットのリストは、ナビゲーションバーではなくメニューを介してアクセスされる。ナビゲーションスタックは目に見えるコントロールとしては表示されないが、ハーフVGAハンドヘルドPCテーマの場合のように、ツールバー上の逆矢印を介して操作可能である。カレントタイルセットのメインタイルはタブストリップの左端のタブとして示されており、サブタイルが右側に続く。

【0101】1.3 ハンドヘルドPC(フルVGAサイズ)

フルVGA(縦640ピクセル、横480ピクセル)スクリーンレイアウトを使用することにより、より多くの情報を同時にスクリーン上に表示可能である。図11に

は外観が示されている。従って、タブストリップを使用してタイルを整理するのではなく、タイルを同時にスクリーン上の1つ又は複数の固定された場所ないし“スロット”に表示する。

【0102】詳細には、総称的UIモデルのフルVGAテーマに対するマッピングは、ハーフVGAテーマの場合と類似する。すなわち、ステップ(a)~(c)は同じである。

【0103】ステップ(d)は以下のように置換される。

【0104】スクリーンが4個のスロットと呼ばれる領域に分割される;付加的属性であるターゲットフレームが、個々のタイルセットのメインタイル及びサブタイル上で定義される。この属性はタイルがスクリーン上で占めるべきスロットのセットを指定する;タイルセットの最初の表示において、タイルセットのタイルが順番に検査される。タイルに対してターゲットフレーム属性内で指定されたスロットが空きである場合、タイルはそれらのスロットに表示され、それ以外の場合は表示されない。

【0105】いずれのスロットも占めないタイルは、スクリーンの底部のオーバフロー領域でキャプションとして最小の形で表示される。このキャプションの1つをクリックすることにより、ユーザは強制的に関連タイルをそのターゲットフレーム属性内に指定されたスロットにおいて表示し、関連キャプションをオーバフロー領域から除外する;タイルを前記ユーザ操作により表示したい場合は、該タイルが占めるべきスクリーンスロットが、現在表示されている1つ又は複数のタイルから除外される;そのようなタイルが、必要とする最小限のタイルセットを持たない場合、該タイルは全てのスロットを解放し、自身をオーバフロー領域におけるキャプションとして表示する;そうでない場合、前記タイルはその表示サイズを縮小し、現在割り当てられているスロットに収められる;タイル上に表示されたUI要素は、該タイルに割り当てられたスロットの数に応じて変わる。例えば、リストビュータイル(List View Tile)の列は関連重要性属性を有し、それによって、利用可能なスペースに納めるために抑制すべき列が決定される。図11の例では、上部のカスタマリストが半分に縮小されているが、それは、そのスロット割り当ての右半分が、カスタマ詳細タイルにより同時に選択されたことに対応した結果である。

【0106】以下の表2は、MDFアプリケーション言語で定義された、シンプルなサービス管理デモンストレーションプログラムを示す。

【0107】

【表7】

以下は、MDFアプリケーション言語により定義された、シンプルなサービスマネジメントデモンストレーションプログラムである。

```
<GUI xmlns="x-schema:ApplicationSchema.xml" title="Test
GUI Definition">
  <DataModel>
    <!-- Code Tables -->
    <Table idTable="t:SN_Status" dsn="SN_Status">
      <Field idField="t:SN_Status.Code" type="string"/>
      <Field idField="t:SN_Status.Text" type="string"/>
    </Table>
    <Table idTable="t:Equip_Type" dsn="Equip_Type">
      <Field idField="t:Equip_Type.Code" type="string"/>
      <Field idField="t:Equip_Type.Text" type="string"/>
    </Table>
    <Table idTable="t:WR_Act" dsn="WR_Act">
      <Field idField="t:WR_Act.Code" type="string"/>
      <Field idField="t:WR_Act.Text" type="string"/>
    </Table>
    <!-- BEGIN: TABLES for synchronization -->
    <Table idTable="t:SN" dsn="SN">
      <Field idField="t:SN.NoteNum" defaultLabel="Request
      id" type="string" length="8"/>
      <Field idField="t:SN.Priority" defaultLabel="Priority"
      type="string" length="1"/>
      <Field idField="t:SN.NoteDate"
      defaultLabel="Received" type="date" length="8"/>
      <Field idField="t:SN.CustNum"
      defaultLabel="Customer Id" type="string" length="9"/>
      <Field idField="t:SN.SerialNum"
      defaultLabel="Installation Id" type="string" length="16"/>
      <Field idField="t:SN.EquipmentNum"
      defaultLabel="Equipment Number" type="string" length="18"/>
      <Field idField="t:SN.ShortTx"
      defaultLabel="Description" type="string" length="20"/>
      <Field idField="t:SN.LongTx" defaultLabel="Memo"
      type="string" length="80"/>
      <Field idField="t:SN.Status" defaultLabel="Completed"
      type="string" length="1"/>
      <Field idField="t:SN.ContactName"
      defaultLabel="Contact" type="string" length="30"/>
      <!-- Not in Jason's model -->
      <Field idField="t:SN.ModelNo" defaultLabel="Model"
      type="string" length="20"/>
    </Table>
    <Table idTable="t:Equip" dsn="Equip">
      <Field idField="t:Equip.EquipmentNum"
      defaultLabel="Equipment Number" type="string" length="18"/>
      <Field idField="t:Equip.InstallDate"
      defaultLabel="Installed" type="date" length="8"/>
      <Field idField="t:Equip.EquipmentTyp"
      defaultLabel="Equipment Type" type="string" length="1"/>
      <Field idField="t:Equip.ModelNo"
      defaultLabel="Model" type="string" length="20"/>
      <Field idField="t:Equip.CustNum"
      defaultLabel="Customer Id" type="string" length="9"/>
      <!-- Not in Jason's model. Submitted by
      EquipmentMem -->
      <Field idField="t:Equip.Serial" defaultLabel="Serial No"
      type="integer" length="18"/>
    </Table>
    <Table idTable="t:Cust" dsn="Cust">
      <Field idField="t:Cust.CustNum"
      defaultLabel="Customer Id" type="string" length="9"/>
      <Field idField="t:Cust.Name" defaultLabel="Name"
      type="string" length="15"/>
      <Field idField="t:Cust.City" defaultLabel="City"
      type="string" length="10"/>
      <Field idField="t:Cust.PostalCode" defaultLabel="Post
      type="string" length="10"/>
      <!-- should add validation code here -->
      <Field idField="t:Cust.Telephone" defaultLabel="Tel"
      type="string" length="15"/>
      <!-- Not in Jason's model, but required -->
      <Field idField="t:Cust.Addr1" defaultLabel="Addr1"
      type="string" length="15"/>
      <Field idField="t:Cust.Addr2" defaultLabel="Addr2"
      type="string" length="18"/>
      <Field idField="t:Cust.Region" defaultLabel="Region"
      type="string" length="5"/>
      <!-- Index refField="t:Cust.Name"/>
      <!-- Index refField="t:Cust.City"/>
      <!-- Index refField="t:Cust.PostalCode"/>
    </Table>
    <!-- WorkReport: check different from t:WorkReport, add
    relationships as required -->
    <Table idTable="t:WorkReport" dsn="WorkReport">
      <Field idField="t:WR.Id" defaultLabel="Report Id"
      type="string" length="12"/>
      <Field idField="t:WR.NoteNum"
      defaultLabel="Notification" type="string" length="8"/>
      <Field idField="t:WR.ActName"
      defaultLabel="Activity" type="string" length="2"/>
      <Field idField="t:WR.ActCode" defaultLabel="Activity
      Code" type="string" length="4"/>
      <Field idField="t:WR.ActTx" defaultLabel="Memo"
      type="string" length="80"/>
      <Field idField="t:WR.ActDate" defaultLabel="Date"
      type="date" length="8"/>
      <Field idField="t:WR.ActTime" defaultLabel="Time"
      type="time" length="8"/>
      <Field idField="t:WR.RequiredPart" defaultLabel="Part
      No" type="string" length="10"/>
      <Field idField="t:WR.PartDelivery"
      defaultLabel="Delivery date" type="date" length="8"/>
      <Field idField="t:WR.EquipmentNum"
      defaultLabel="Equipment Id" type="string" length="18"/>
      <Field idField="t:WR.CustNum"
      defaultLabel="Customer Id" type="string" length="9"/>
    </Table>
    <!-- END: TABLES for synchronization -->
    <Table idTable="t:Model" dsn="Model">
      <Field idField="t:Model.ModelNo" defaultLabel="Model
      No." type="string" length="20"/>
      <Field idField="t:Model.Manufacturer"
      defaultLabel="Manufacturer" type="string" length="15"/>
      <Field idField="t:Model.Description"
      defaultLabel="Description" type="string" length="15"/>
    </Table>
    <Table idTable="t:Symp" dsn="Symp">
      <Field idField="t:Symp.Id" defaultLabel="Id"
      type="string" length="2"/>
      <Field idField="t:Symp.Description"
      defaultLabel="Description" type="string" length="20"/>
      <Field idField="t:Symp.ModelNo" defaultLabel="Model"
      type="string" length="20"/>
    </Table>
    <Table idTable="t:Check" dsn="Check">
      <Field idField="t:Check.Id" defaultLabel="Id"
      type="string" length="3"/>
      <Field idField="t:Check.Likelihood"
      defaultLabel="Likelihood" type="string" length="3"/>
      <Field idField="t:Check.Description"
      defaultLabel="Description" type="string" length="15"/>
    </Table>
    <Table idTable="t:Part" dsn="Part">
      <Field idField="t:Part.PartNo" defaultLabel="Material"
      type="string" length="10"/>
      <Field idField="t:Part.Text" defaultLabel="Description"
      type="string" length="15"/>
    </Table>
    <Relation id="reln:Cust_SN" table1="t:Cust"
    field1="t:Cust.CustNum" table2="t:SN" field2="t:SN.CustNum"
    cardinality1="1" cardinality2="n"/>
    <Relation id="reln:WR_SN" table1="t:WorkReport"
    field1="t:WR.NoteNum" table2="t:SN" field2="t:SN.NoteNum"
    cardinality1="n" cardinality2="1"/>
    <Relation id="reln:Model_Symp" table1="t:Model"
    field1="t:Model.ModelNo" table2="t:Symp"
    field2="t:Symp.ModelNo" cardinality1="1" cardinality2="n"/>
    <Relation id="reln:Cust_Equip" table1="t:Equip"
    field1="t:Equip.CustNum" table2="t:Cust"
    field2="t:Cust.CustNum" cardinality1="n" cardinality2="1"/>
  </DataModel>
</GUI>
```

【表9】

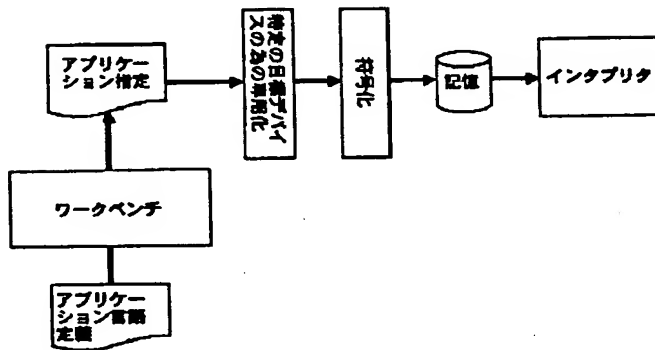
【図面の簡単な説明】

【図7】 図7は、インタプリタアーキテクチャの概略で

【図11】図11は、表2のMDFアプリケーション用の480×640ピクセルハンドヘルドPCレイアウトテーマを例示する、コンピュータスクリーンの一連のビューである。

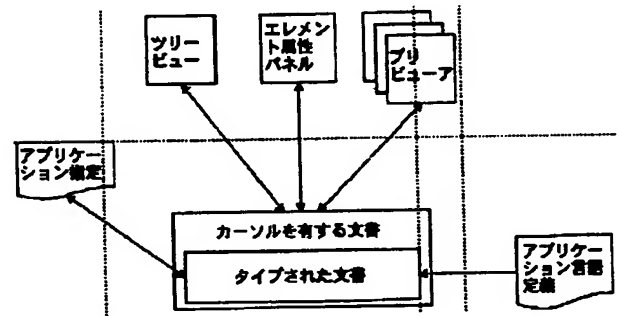
【図12】 図12は、表2のMDFアプリケーション用のサーバ側インタプリタのWAPテーマを例示する、マイクロブラウザ装備無線電話のディスプレイの一連のビューを示す図である。

【図1】

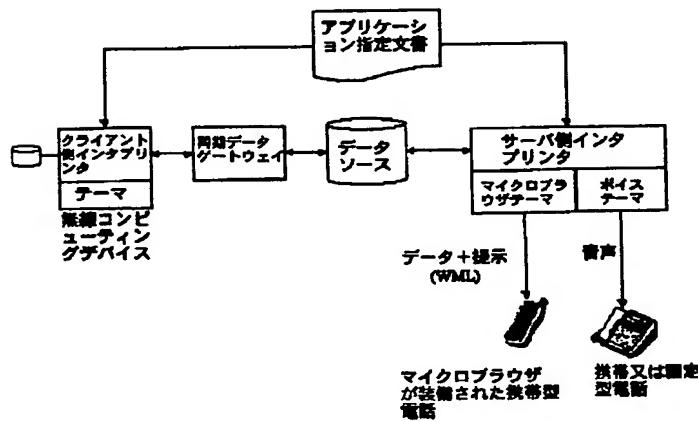


【図13】 図13は、表2のMDFアプリケーションを実行しているユーザと、サーバ側インタプリタの音声テーマ間の電話上で行われる対話の記録である。

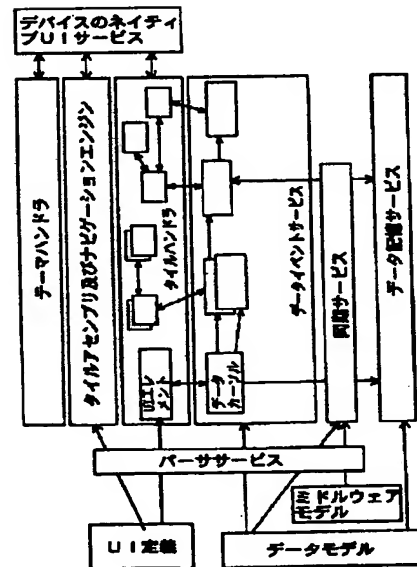
【図3】



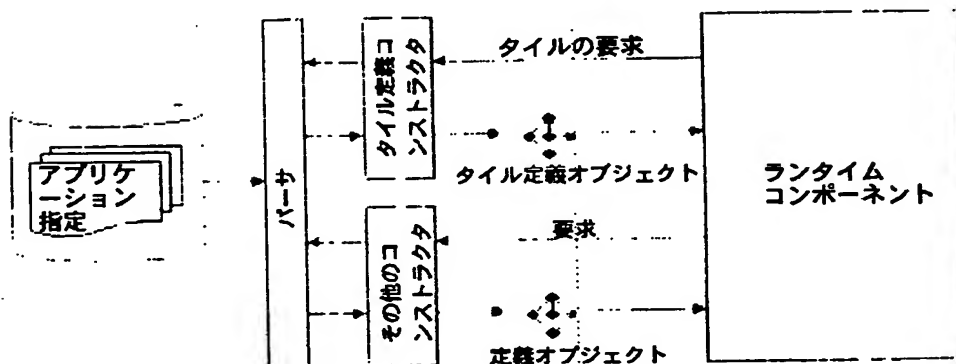
【図2】



【図7】



【図8】



【図4】

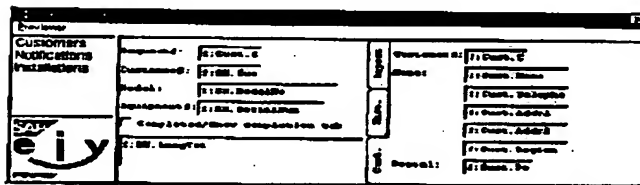
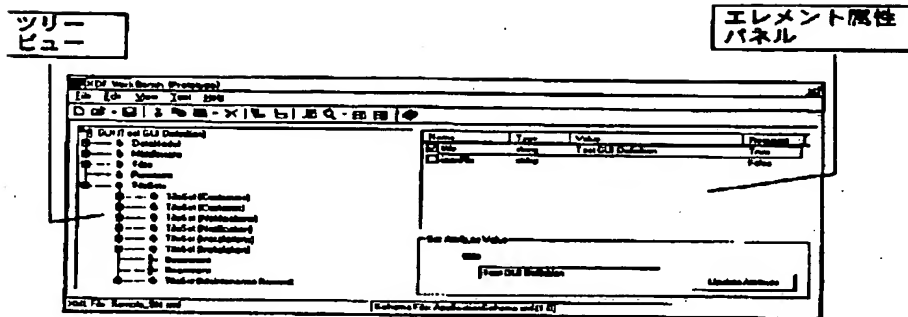


図4a ハンドヘルドPCプリビュー

図4a パームPCプリビュー

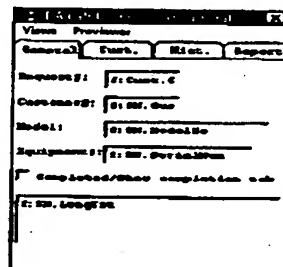
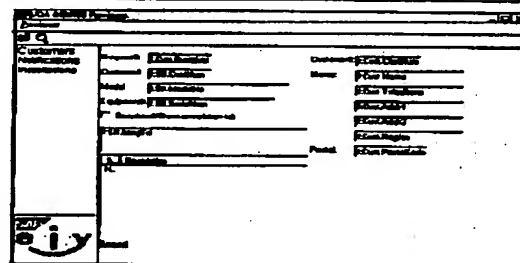
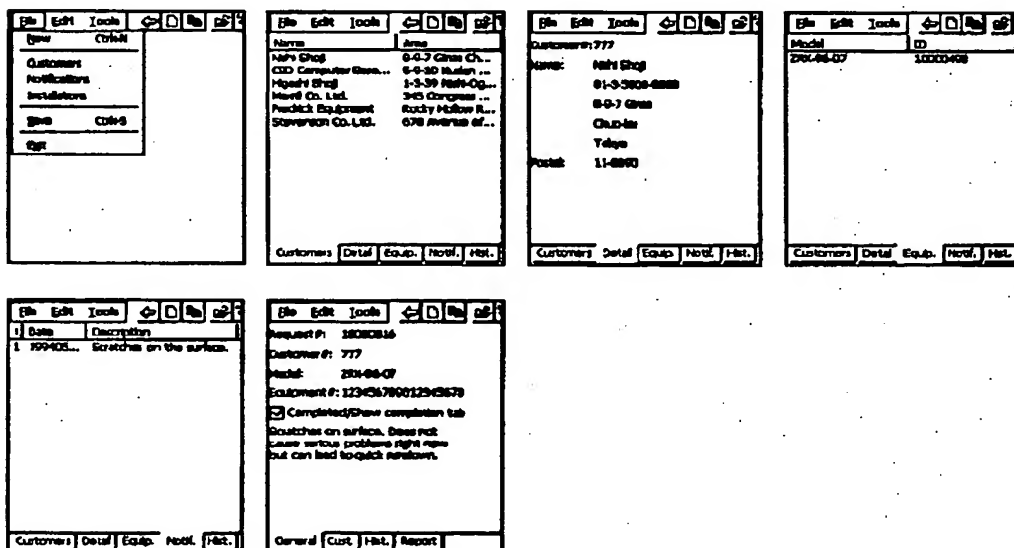


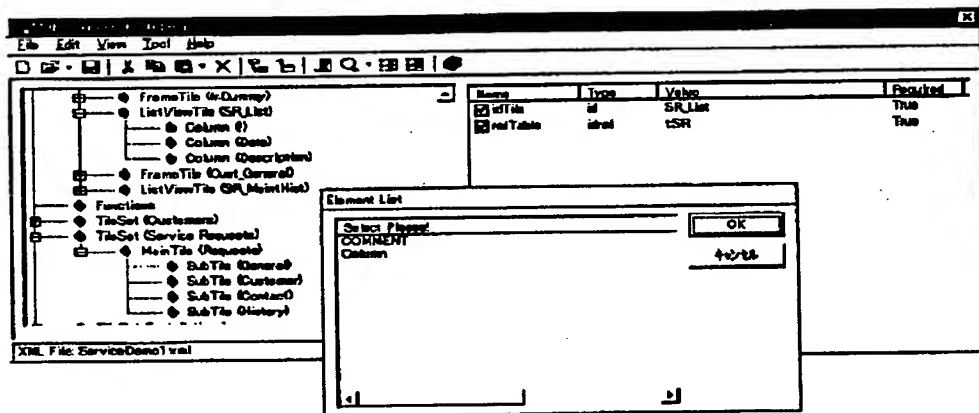
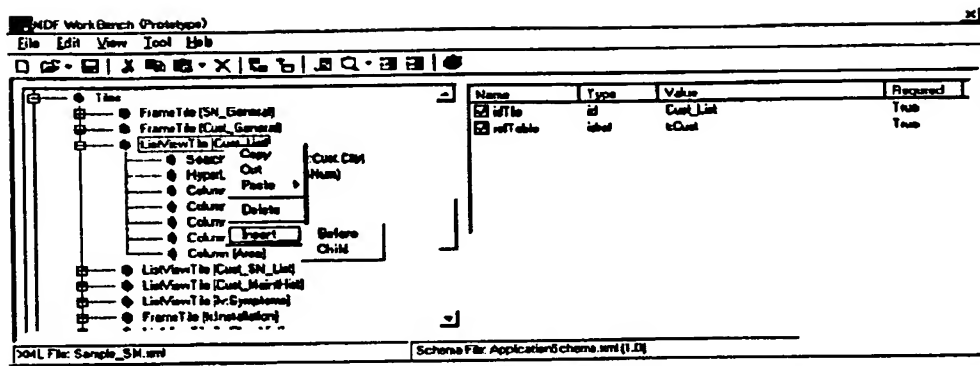
図4a VGAサイズハンドヘルドPCプリビュー



【図10】



【図5】



【図6】

<AttributeType name='cardinality1' dt:type='enumeration' dt:values='1 n'>

Name	Type	Value	Required
table2	idref	ESN	True
field1	idref	ESN.CustNum	True
field2	idref	ESN.CustNum	True
cardinality1	enumeration	1	True
cardinality2	enumeration	n	True
dependentOn...	enumeration	neither	False

Set Attribute Value

cardinality1

Update Attribute

XML File: Sample_SM.xml Schema File: ApplicationSchema.xml (1.0)

Name	Type	Value	Required
table2	idref	ESN	True
field1	idref	ESN.CustNum	True
field2	idref	ESN.CustNum	True
cardinality1	enumeration	1	True
cardinality2	enumeration	n	True
dependentOn...	enumeration	neither	False

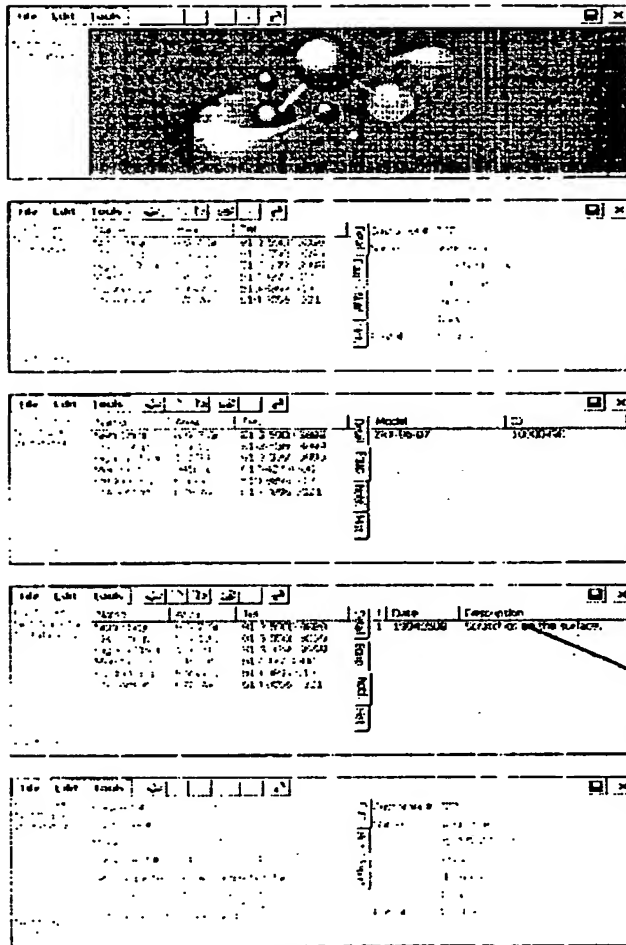
Set Attribute Value

field2

Update Attribute

XML File: Sample_SM.xml Schema File: Application

【図9】



このハイパーリンクを実行することにより、通知(Notification)タイルセットが作成され、スタックの最上部に付加される。

【図11】

Figure 11 displays three screenshots of a software interface, likely a customer management system. The top-left screenshot shows a 'Customers' list with columns for ID, Name, Age, Region, Postal, and Tel. The top-right screenshot shows a 'Customers' list with columns for Name, Age, and Tel. The bottom-left screenshot shows a 'Customers' list with columns for Name, Age, and Tel, and a 'Customers' list with columns for Name, Age, and Tel.

【図13】

ダイアログの写し

ハンドヘルドPCテーマ (比較用)

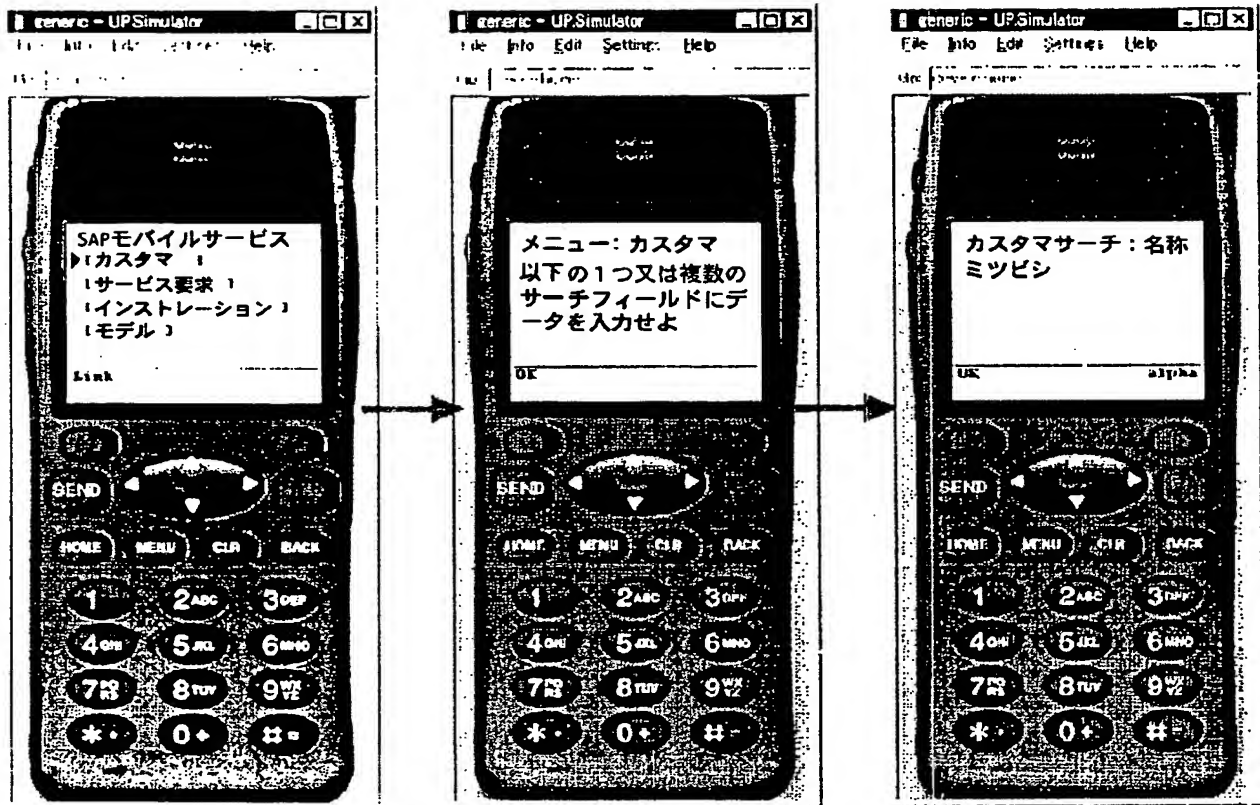
- ① 電話セールスフォースオートメーションシステムへようこそ。メインメニューからアイテムを選んでください。メニューにはカスタマ、コンタクト、及び活動があります。
- ② (選択) カスタマ。
- ③ 選択されたアイテムはカスタマです。5人のカスタマが存在します。どうしますか? リストを眺み上げるよう要求するか、又は特定のアイテムに関して名前で照会することができます。
- ④ Wでサーチ。
- ⑤ 照会の結果は以下の通りです: 2人のカスタマが見つかりました。アイテム1: カリフォルニアのBarlow Hospital。アイテム2: ...アイテム1とアイテム2のどちらに興味がありますか?
- ⑥ アイテム1。

Figure 13 displays a screenshot of a software interface showing a list of customers. The list includes columns for Name, State, and City. The customers listed are Barlow Hospital, St. Mary's Health Center, Pacific General Hospital, Pacific University Hospital, and Wilson Healthcare Assoc. The interface also shows a 'Customers' list with columns for Name, Age, and Tel.

- ① 以下は選択されたカスタマの一覧です: カスタマ名はBarlow Hospital...eメールアドレスはwebmaster@barlow.com.です。どうしますか? このカスタマのコンタクト又はコンタクト履歴に関する情報を得ることができます。また、このカスタマに関してより多くの情報を得ることもできます。又は、メインメニューに戻ったり、今聞いたことを繰り返すよう要求することもできます。
- ② カスタマ詳細をチェック。

Figure 13 displays a screenshot of a software interface showing a list of customers. The list includes columns for Name, State, and City. The customers listed are Barlow Hospital, St. Mary's Health Center, Pacific General Hospital, Pacific University Hospital, and Wilson Healthcare Assoc. The interface also shows a 'Customers' list with columns for Name, Age, and Tel.

【図12】



【外国語明細書】**1. Title of Invention**

A METHOD FOR DEFINING CLASSES OF DATA-CENTRIC APPLICATIONS

2. Claims

1. A method for defining classes of data-centric applications that share particular generic user interface models comprising the steps of:

Determining the types of distinguished User Interface elements called Tiles, and their constituent elements, required for said generic User Interface Model;

determining the cardinality and relationships between the Tiles in groups of Tiles called Assemblies;

determining the logical model for Navigation between the Assemblies, and the required user operations on the model state.

2. The method for defining an application specification language in a meta language suitable to each said class of applications comprising the steps of:

defining elements which specify the Tile types and constituent elements, the containment relationships between the elements, and the attributes required to express any referential relationships between the elements;

defining the attributes on the elements required to express the properties of the Tile and User Interface elements;

defining any attributes on the elements required by the Navigation model.

3. The method for creating an application system for each said class of applications, for development and execution of instances of said class, comprising the steps of:

Customizing a generic Workbench with event handlers that realize constraints on the validity of application specifications which are not representable in the meta-language.

Customizing a generic Interpreter, comprising the steps of:

a Creating object models for the specifications of Tiles, User Interface

elements, and Assemblies;

b implementing definition object constructors that will create object model instances from application specifications or fragments thereof;

c implementing the creation of UI Model instances from definition objects, in a suitable computer programming language.

4. The method of creating applications of a particular class, comprising the steps of:

creating a document containing the specification of said application using the Workbench component of the Application System;

optionally specializing the said application specification document for the particular device;

encoding the application specification document into a form suited for the Interpreter.

5. The method of providing multiple concrete user interfaces

corresponding to each said generic user interface model, comprising the steps of:

defining a representation on the output media of the device for each element of the UI Model;

defining the concrete user-interface for each operation on the UI Model;

defining additional elements and attributes in the Application

Specification Language as required by the output representation and the UI Model operation user-interface;

implementing each Theme as a component of the Interpreter in a suitable computer language;

providing a mechanism that allows the Interpreter to select an appropriate Theme suitable for the particular computing device on which the user interface is presented;

6. The method of providing online, offline and mixed operation of said applications on a mobile device by presenting to the user the appearance

of a response-reply style of interaction with the remote system;

7. The method of creating variant interpreters for specific environments.

3. Detailed Explanation of the Invention

1 Field of the Invention

The present invention relates generally to the creation of data-centric applications for computing devices, and more particularly to a scheme for defining classes of applications sharing common user-interface models, a technique for the presentation of the user-interface models in a manner suitable for various computing devices of different user-interface capabilities, sizes and shapes, and a technique for rapid creation of conformant applications which can be used in online, occasionally connected, or offline modes.

2 Background of the Invention

2.1 The problem

Recent technology developments have created an opportunity to field low-cost, low-maintenance small mobile ("pervasive") devices to professional and non-professional workers who are not always, or perhaps never, within reach of a desktop computer. These pervasive devices must be able to integrate with corporate data systems and IT-enabled business processes. Furthermore, the current state of communication reliability, communication costs, and device battery capacity make it necessary to support disconnected as well as connected modes of usage. Unfortunately, there is a dearth of tools to meet this pair of requirements.

From an application development, maintenance and deployment standpoint, there is the additional problem of supporting a multiplicity of device form factors — for example Windows CE consumer devices are

available in at least 4 separate screen sizes. Micro-browser equipped wireless telephones are also receiving a great deal of attention. Finally, the most pervasive of all devices is the traditional telephone; voice medium also can deliver corporate content via Interactive Voice Response systems using telephone push buttons or voice-recognition/synthesis interfaces. Developers face severe learning ramps in delivering solutions for this wide array of unfamiliar devices.

2.2 The task

Provide a way to rapidly create customizable applications for pervasive computing devices. These applications must integrate with corporate data systems such as SAP R/3. Further, since communication infrastructures are developing both rapidly and unevenly throughout the world, the development method must support mixed online and offline usage of the developed applications without imposing large discontinuities and resultant training costs on end-users.

The requirements for application logic on these devices is usually light because the portions of the business processes executed on the devices are relatively simple, and because the devices themselves have limited capacity.

3 Summary of the Invention

The invention relates to a scheme for rapid creation of data-centric applications which can be deployed on many different types of computing devices.

Applications are specified at a high-level, optionally specialized for a particular media or form-factor, and executed by an appropriate interpreter. A family of declarative application specification languages is specified for this purpose; these languages are capable of expressing the data-model, integration with corporate data, and an abstracted user-interface. Each language corresponds to a particular generic user-

interface paradigm, and is used to specify applications that fit the paradigm. These user-interface paradigms are mapped on to different devices in a way appropriate to the capabilities of each device and media. Interpreters for each language interpret the application specification and present the user interface. The interpreters provide the basic user-interface framework (navigation model, binding of data to UI elements), data-services (validation, inter-object relationships and navigation, data event control, capture of data changes), and middleware-services (data synchronization, virtual request-reply). The built-in functionality can be augmented by using of common scripting languages and by the use of custom object components based on COM or similar technologies.

The integration between the client device and the server is realized by a data synchronization mechanism built in to the interpreters. Therefore, the basic mode of operation is offline. However, mixed offline and online application usage is supported by the interpreters' additional capability of simulating the "request-response" style of interaction users expect of an online connection. The interpreters can present this appearance by ensuring that data-synchronization is done in near real-time, and that the data changes resulting from the synchronization are reflected on the user-interface without explicit user action as the results become available. The former condition can be realized by standard messaging and communication services of the computing devices. The latter condition is fulfilled by constructing the interpreter in such a way that the data synchronization layer has suitable hooks into the user-interface data event layer, such that server-initiated data changes are automatically propagated to the UI elements.

It is an object of this invention to define a framework by which classes of applications sharing a common user-interface model are defined.

It is a further object of this invention to define a process and method for

creating an application development system for each particular user-interface model.

It is a further object of this invention to provide a method of defining computer languages used to specify particular applications of the user-interface model.

It is a further object of this invention to provide interpreters for certain computing devices that realize the particular user-interface model.

It is a further object of this invention to provide a layout component in the interpreters that realizes the particular user-interface model in a manner suited to each computing device.

It is a further object of this invention to provide for offline, online and mixed usage of the applications with the same user interface model and realization.

It is a further object of this invention to provide functionality in the interpreters for certain computing devices for synchronizing data between the device and one or more servers.

It is a further object of this invention to provide functionality in the interpreters for certain computing devices for reflecting changes in the data initiated by a server in the user-interface, without explicit user-action.

It is a further object of this invention to provide techniques for improving the performance of these interpreters on small computing devices such as Personal Digital Assistants.

It is a further object of this invention to define a process and method for creating applications compliant with the particular user-interface model.

4 Prior Art

4.1 Server-based layout to support multiple device form factors and media

Pertinent to claim (a)

These solutions focus on online usage; in some cases, it is possible to extend these solutions to limited offline scenarios by use of offline browsing. The latter technique has many well-known weakness related to presentation flexibility (dynamically selecting and formatting data), data validation, data volume scalability (support for delta synchronization, etc.).

4.2 Form languages

Pertinent to claim (c)

These languages focus on the User Interface only; an example being Celesta SmartForms. They support only simple data models and weak integration with corporate data.

4.3 Voice proxies

Pertinent to claim (a)

These gateway services attempt to convert preexisting Web pages dynamically a form suitable for telephones by the use of voice synthesis and speech recognition. These proxies must infer and impose structure on the Web pages in order to present the large amount of visual information on a low-bandwidth voice interface.

With the current state of the art, this approach is limited in its applicability and robustness.

5 Application Model

5.1 Modeling applications

We think of an application as a combination of several related components, in accordance with the following model:

Application = Data-Model + Middleware-Model + Business Logic + User Interface

The Application Specification Languages that we define allow us to address each of these components, although it is not necessary for all of

these components to be present. The Data-Model defines the local data on which the application will act: a set of tables and their inter-relationships. The Middleware-Model defines the content of the messages exchanged between the client device and servers to accomplish data synchronization, and the relationship between the messages and the objects of the Data Model. The Business Logic component comprises the pool of functions and the scripts associated with various application events. The User Interface defines the means by which the user will interact with the application.

We further factor the User Interface component as follows:

User Interface = User Interface Model + Theme

User Interface Model = Tiles + Assemblies + Navigation

where

Tile = a view of an application-level object, such as customer, contact, etc.

A set of interface elements presented together.

Assembly = a group of Tiles and their interrelationships;

E.g. a group of views into a set of related business objects.

Navigation = model of the state of the User Interface Model, and the operations on the UI Model, including access, instantiation and transition between Tiles and Assemblies; the Navigation state may include memory of previous states.

Theme = presentation of the User Interface Model on a particular device-type. Realization of a Theme may require additional "state" information to represent the status of the presentation at runtime; this information is logically associated with and extends the Navigation state.

The above equations define a family of Application models parameterized by the concrete definitions of the Tile, Assembly and Navigation models.

The particular Tile, Assembly and Navigation models we choose give us

the requirements for the Application Specification Language that specifies applications complying with the corresponding UI model. The resulting applications are presented on different devices and media in accordance with a Theme suitable for each particular device and medium type.

In the rest of this section, we present several User Interface Models.

5.2 MDF (Tile-set Stack Model)

The User Interface Model for the MDF framework, as is the case for all of the Application Specification Languages, is characterized by the definitions of the Tile, Assembly and Navigation models. For MDF, we have

Tile = Set of UI elements that are bound to fields within a particular Business object type. For example, a Customer Detail Tile will bind to the collection of Customer, and individual UI elements to fields such as Customer Number, Name, Industry type, etc.

Assembly = A set of Tiles, called a Tile Set, organized in a hierarchy (or collection of hierarchies). One or more Tiles are designated as "Main" Tiles, each of which is the root of a tree of "Sub" Tiles. A Sub-Tile is related to its parent by a query on the set of all data items associated with the Sub Tile, where the query is parameterized by properties of the currently selected item in the parent. This allows, for example, the selection of a Customer in the Main Tile to cause the automatic selection and display of the related Sales Orders in a Sub Tile.

Navigation = The Navigation state comprises a stack of Assembly (e.g. Tile Set) instances, of which the top is said to be current; only the current Tile Set is displayed and can be interacted with by the user. Logically, all Tiles are presented to the user simultaneously; however the layout Theme for a particular device may require the user to explicit perform some action to access the individual tiles. Such actions do not affect the navigation stack. A new Tile Set instance can be created by executing a

"Hyperlink" in one of the Tiles, in which case the new instance is pushed onto the top of the stack. This operation can be reversed by discarding the top of the stack. Some Tile Sets are distinguished as top-level Tile Sets; there is a mechanism for creating instances of such Tile Sets directly, in which case the stack is cleared prior to creating the new instance.

This User Interface Model, and the corresponding application language, is a preferred embodiment. See Table1 for the Application Specification Language for this model. See section 8.3.1 for examples of MDF Themes.

5.3 Pervasive Workflow (Dialog Model)

The "Dialog" UI Model is suited for simple Workflow applications, where the end-user is required to make a simple choice based on a small amount of predetermined data at a particular point in the Workflow; this choice step can be presented on a mobile computing device. A User Interface Model suitable for this scenario is described by the following definitions:

Tile - Set of UI elements that are bound to fields on the Workflow container. For example, the workflow container for Vacation Request Approval would have UI elements bound to the Requestor, Date, Reason and other fields.

Assembly - An assembly comprises a single Tile. For simplicity, in the rest of this section we do not distinguish between the Tile and the Assembly that it comprises.

Navigation - The Navigation state comprises a single Tile instance, designated as the current Tile. The user can only interact with this Tile. All Tile instances are available in a pool, which the user can access in order to designate a Tile instance as current. The user may return an instance back to the pool by selecting another instance from the pool. There is a mechanism for the User to indicate that processing of the Tile instance is complete, whereupon the Tile is logically removed from the pool. The

user may also initiate a new Tile instance and add it to the pool, by selecting from the list of Tiles defined in the application.

This UI Model can be elaborated to manage a Tile instance pool of more complexity; for example, the Workflow represented by each Tile instance could have an associated priority or additional states.

6 Application Systems

6.1 Overview

An Application System provides the mechanisms by which applications sharing a particular User Interface Model are created.

An Application System comprises an Application Specification Language, a Workbench, and an Interpreter. The Application Specification Language defines the language elements required to express applications in the User Interface Model. The Workbench is the component that the application developer uses to create and assemble instances of the language elements to form the application specification. The Interpreter is the component that executes the application specification and presents the user interface on the computing device.

The steps to create an Application System comprise the following steps. The first step is to define the User Interface Model, comprising the steps of:

- determining the types and functionality of the Tiles and any User Interface elements contained on the Tiles;
- determining the cardinality and relationships between the Tiles in the Assemblies, and any particular requirements for operations between Tiles in certain specialized Assemblies;
- determining the logical Navigation model, and the required user operations on the model state.

The second step is to define in a meta-language the Application Specification Language, comprising the steps of:

defining elements which specify the Tile types, the User Interface elements if any contained in the Tiles, the containment relationships between the elements, and the attributes required to express any referential relationships between the elements;

defining the attributes on the elements required to express the properties of the Tile and User Interface elements;

defining any attributes on the elements required by the Navigation model.

The third step is to customize the generic Workbench, comprising the steps of:

creating event handlers that realize constraints on the validity of application specifications which are not representable in the meta-language.

The fourth step is to define the Layout themes for each supported device, comprising for each device the steps of:

defining a representation on the output media of the device for each element of the UI Model;

defining the user-interface for each operation on the UI Model;

defining additional elements and attributes in the Application Specification Language as required by the output representation and the UI Model operation user-interface.

The fifth step is to realize the UI Model of the Interpreter, comprising the steps of:

Creating object models for the specifications of Tiles, User Interface elements, and Assemblies; such objects termed "definition objects";

implementing definition object constructors that will create object model instances from application specifications or fragments thereof;

implementing the creation of UI Model instances from definition objects, in a suitable computer programming language.

The sixth step is to realize the Themes in the interpreter, comprising for

each Theme the steps of:

implementing each Theme as a Layout handler in a suitable computer language;

providing a mechanism that allows the Interpreter to select the appropriate Theme for the computing device.

Details of this process are given in section 8 where a preferred embodiment, MDF, is described.

A preferred embodiment of the syntax of the Application Specification Language is the eXtensible Markup Language (XML, World Wide Web Consortium, www.w3c.org). The use of the term "XML" in this context refers to the World Wide Web Consortium specification of XML, as well as encoding schemes for XML into binary or other forms that preserve the XML structure.

Another preferred embodiment of said syntax is a binary encoding of XML, "WAP Binary XML encoding" (WAP Consortium, www.wapforum.org).

A preferred embodiment of the meta-language for defining the Application Specification Language is XML Schema (World Wide Web Consortium, www.w3c.org).

6.2 Workbench

The Workbench is the tool with which application developers create applications. Its operation is directed by a document containing the definition of an Application Specification Language; the Workbench constrains its document browsing and editing functions to comply with said language. Notwithstanding the above, some customization of the Workbench may be required to realize document validity criteria not adequately expressed by the meta-language in which the Application Specification Language is defined.

6.2.1 Architecture

FIG. 3 shows a simplified object model of the Workbench. The Workbench comprises a Typed Document and associated Document Cursor, a TreeView and Element Attribute Panel for browsing and editing the Typed Document, and a number of Previewers for simulating during the development process the final presentation of the document.

The Typed Document component uses the Application Specification Language definition to understand the application specification document structure and to determine the allowable edit operations on each element and attribute.

The Typed Document component provides Application Programming Interfaces (API) to load the Application Specification language, to create, load and save application specification documents, and to provide information on valid edit operations to the other components to guide their proper operation. Any changes to the application specification document will be broadcast to all other components, which will update their internal state and displays accordingly.

The notion of "current element" is realized by the Document Cursor. This component contains a reference to a distinguished element in the application document designated as the current element. Any of the components, including the TreeView and Previewers, which are capable of browsing the document elements, can update the current element reference by calling a Document Cursor API. The Document Cursor then broadcasts the new reference to all the other components, which update their internal state and screen displays accordingly.

The TreeView displays the document as a hierarchy mirroring the actual structure of the document. Context-sensitive menus are provided for deletion and insertion of elements into the document.

The Element Attribute Panel displays the set of attributes defined for the current element. The attributes are represented on the Panel in a form

suited to their type and other meta-attributes.

The Previewer components present the application document in a form that approximates how the application would be presented if executed by the Interpreter. Each Previewer component realizes a different concrete presentation (Theme), and can be started separately through the Workbench user interface.

The current element, if said element corresponds to a User Interface Element, will be visible and highlighted in each open Previewer. Also, selecting the representation of a User Interface Element in a Previewer will cause that element to become the current element; as a result, the TreeView and all Previewers will update their displays accordingly.

6.2.2 Operation

The operation of the Workbench is described in detail in section 7.2.

6.3 Interpreter

The Interpreter is the component that executes the application specification and presents the user-interface.

6.3.1 Architecture

The architecture of the Interpreter is presented in FIG. 7. Of particular note is the subsystem entitled "XML Services". This subsystem provides access to the application specification, separated for didactic purposes into the GUI Definition, Data Model and Middleware Model components shown on the far left. At the far right is the Device Native UI Services. These are the user interface services provided by the particular computing device, including such things as window management for a device with a graphical display or speech synthesis for a device with a voice interface. The other components are collectively named the Runtime components.

- The Tile Handler component realizes the collection of Tile types, and any User Interface elements that a Tile may contain. Tile instances, and any contained UI elements, are created and destroyed in this component

as directed by the Tile Assembly and Navigation Engine component.

- The Tile Assembly and Navigation Engine component realizes the generic User Interface Model. During the creation of an Assembly, this component creates the inter-Tile relationships by setting up an data event propagation network formed by UI elements and Data Cursors.
- The Theme Handler component provides the mapping of the generic User Interface Model realized by the Tile Assembly and Navigation Engine component to a concrete user interface. The Theme Handler acts as the bridge between the Tile Assembly and Navigation Engine on one side and the user on the other. If required by the defined Theme, the Handler may invoke Show or Hide operations on the individual Tiles, or equivalents for a non-graphical user interface.
- The Data Event services component provides the glue that ties the individual Tiles, UI elements, and Synchronization services together. Data Cursor objects in this component realize the notion of a sequence of Objects and a distinguished position in the sequence; the distinguished position is called a "cursor" and the Object at that position is designated as the "current object". Objects in the Data Event services component propagate events relating to the data-store among the Data Cursors and UI elements.
- The Synchronization services component is responsible for the synchronization of the data-stores between the mobile client device and the remote server. The component intercepts changes to the Data-store originating from the Tile Handler component, records and assembles them into messages which are eventually sent to the server. In the other direction, the Synchronization services component receives changes from the server in the form of messages from the server requesting said changes, and applies them to the Data-store; the receiving action is completed by issuing data events into the Data event services component

corresponding to the changes.

- The Data-store services component encapsulates the persistent data-storage of the computing device. Said persistent storage may be a relational database management system, or a simpler facility such as a Windows CE Database (Microsoft Corp, Seattle, WA).

The next section goes into more detail.

6.3.2 Application Document services Component

The Application Document services component is responsible for understanding the encoding of the application specification, and presenting the specification in a form that can be efficiently used by the other components. FIG. 8 illustrates the interpretation of application fragments by the Application Document services component. In this figure, a runtime component requests the specification of a Tile definition, and receives an object representation of the Tile definition in reply.

The generic Parser reads the Tile definition fragment while issuing parser events to a constructor object specific to the element at the root of the fragment, the Tile element. This constructor object creates an object representation of the Tile definition in working memory on the device, and delivers this object representation back to the runtime components of the interpreter.

The preferred embodiment requests the object representation on the first reference to the contents of the fragment, and caches the object representation in case of further use.

6.3.3 Interpreter Variations

An Application System may provide Interpreter variants that are specialized for special environments or system configurations. FIG. 2 illustrates two Interpreter variants.

The client-side Interpreter of FIG. 2 executes completely on the client.

This preferred embodiment is a full realization of the architecture of FIG. 7

The server-side Interpreter of FIG. 2 executes completely on the server, with the exception of the presentation layer. This variant does not require the Synchronization services component. On this Interpreter, the WAP microbrowser Theme realizes a Wireless Markup Language (WAP Consortium, www.wapforum.org) presentation layer using the native microbrowser on the device. The Voice Theme realizes a voice-recognition and voice-synthesis user interface; the presentation layer here is the audio input and output on the telephone.

7 Application Development process

This section describes the process an application developer goes through to create an application.

7.1 Overview of the process

FIG. 1 shows the steps in creating an application. An application is defined in the context, and by usage, of components of an Application System. The Workbench component assists the developer in creating an application specification compliant with the Application Specification Language and thus with the application model, including the specific Pane-, Assembly- and Navigation-specific User Interface Model.

The process of developing an application comprises the following steps:
creating a document containing the specification of said application using the Workbench component of the Application System;
optionally specializing the said application specification document for the particular device;
encoding the application specification document into a form suited for the Interpreter.

The application specification can then be stored in a suitable storage system, from which it is retrieved and executed by the Interpreter upon request.

7.2 Details of the process

An application specification is a XML document that conforms to a particular Application Specification Language. Such languages are specified in a meta-language such as Document Type Definition (DTD) or XML Schema. The Workbench is an XML document editor that assists the application developer in creating the said document by providing document validation and context-sensitive editing compliant with the Application Specification Language. In the preferred embodiment, the XML Schema meta-language is used.

The developer creates a new document by invoking a New command from the Workbench menu. Subsequently, the developer modifies the document by adding or deleting XML elements, or by adding, modifying, or deleting attributes of these elements. Operations on elements are performed through user-interface actions on the Tree View, while operations on element attributes are performed through the Element Attribute Panel.

To insert a new instance of an element into the document, the developer selects an element in the XML TreeView, and invokes a popup-menu. From the menu, he specifies whether the element is to be added at the same level before or after the selected element, or as a child of the selected element. In the example illustrated in FIG. 4, the developer has selected the ListView element and specified the insertion of a child element. The Workbench responds with a dialog box showing a list of the candidate elements at this location of the document; the candidate elements are determined automatically by inspection of the Application Specification Language.

To set the value of an attribute of a particular element, the developer selects the element in the XML TreeView. The Workbench determines the list of all possible attributes for this element by inspection of the Application Specification Language, and displays them in the Element

Attribute Panel. Information on the type and other meta-attributes of each attribute determines its visual appearance on the Attribute Panel. For example, an attribute defined as an enumerated type could be displayed as a dropdown list containing the set of enumerated values, as illustrated in FIG. 6.

The application developer continues in this fashion to create the application specification document. In one preferred embodiment of the process, named "prototyping", the User Interface portion is created first in order to gain end-user understanding and feedback on the design. The Data Model portion is created later. In another preferred embodiment, when the Data Model is known a priori, its portion of the document is created first, followed by the creation of the User Interface portion.

The prototyping embodiment is enabled by the Previewer components, which are capable of approximating the presentation of the application on the Interpreter. A Previewer component may effect this simulation with documents that are incomplete and therefore not capable of execution by the Interpreter itself. Since each Previewer realizes the simulation of a particular layout Theme, and since the Previewers can be displayed simultaneously, the developer may confirm the presentations of the application on each of the target devices at the same time. Since the architecture of the Workbench also provides that any changes to the application document are broadcast to all components, the confirmation of the presentation on each target device can be done while the application is being created or modified.

After completing the creation of the application specification document, the developer uses the validation function of the Workbench to ensure that the Interpreter can correctly execute the document. The validation function is also driven by the language definition.

Finally the document is transformed and encoded (see FIG. 1), and then stored ready for use by the Interpreter.

8 Mobile Data application Framework Application System

In this section, we describe a preferred embodiment of the Application System, the Mobile Data application Framework (MDF) system.

8.1 MDF Application Specification Language

The language corresponding to the User Interface model described in section 5.2 is given in Table1.

Table1 is the MDF Application Specification Language is an XML-compliant language, and is defined by an XML Schema (based on Microsoft realization of the draft W3C standard as of the time of writing).

Table 1

<p><!-- XML Schema Extensibility for MDF schemas --></p> <p><Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:sd="urn:schemas-microsoft-com:sdtypes"></p> <p><!-- Attribute to support subtyping of document krefs. Usage in schema: <AttributeType name="xref" dttype="kref" extrefType="xref" where="yyy" /> where xxx is/are the name(s) of tag(s) to which the attribute can refer, and yyy is/are the name(s) of the tag(s) which are the scope for the reference. Ids will only be matched if lie outside the innermost containing element with nodename from the set of tags yyy. The document processing application is responsible for enforcing the subtyping. --></p> <p><AttributeType name="xrefType" dttype="string"/></p> <p><AttributeType name="xrefScope" dttype="string"/></p> <p><!-- Support layout --></p> <p>displayUnit: attributed to containers that can be shown or hidden</p> <p>mergeable = could be combined with parent elements</p> <p>unmergeable = never combined with parent elements</p> <p><AttributeType name="displayUnit" dttype="enumeration" dtvalues="unmergeable mergeable"/></p> <p><!-- Support for fragmentation --></p> <p><AttributeType name="fragment" dttype="boolean"/></p> <p><!-- Support for translation --></p> <p><AttributeType name="translatable" dttype="boolean"/></p> <p><!-- Descriptive text --></p> <p><AttributeType name="description" dttype="string"/></p> <p><!-- Support for allowing identification of layout-related attributes and tags --></p> <p><AttributeType name="layout" dttype="enumeration" dtvalues="all VGA HVGA PPC"/></p> <p><!-- Temporary extension-attributes, while waiting for Microsoft to upgrade their XML Schema support! --></p> <p><AttributeType name="length" dttype="int"/></p> <p><AttributeType name="minLength" dttype="int"/></p> <p><AttributeType name="maxLength" dttype="int"/></p> <p><!-- Next extension-attribute, if present, overrides dttype --></p> <p><AttributeType name="type" dttype="enumeration" dtvalues="bin hex bin.bee64 filename icon bitmap"/></p>	<p><AttributeType name="fieldType" dttype="enumeration" dtvalues="string number integer float boolean date time datetime enumeration"/></p> <p><Element type="SupportType"></p> <p><attribute type="fieldType" required="yes"/></p> <p></Element type="SupportType"></p> <p></Schema></p> <p><!-- MDF Application Schema, \$Header\$ --></p> <p><Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:sd="urn:schemas-microsoft-com:sdtypes" xmlns:ext="x-schema:SchemaExtension.xml"></p> <p><!-- map_mdf toplevelname=GUI version=1.0? --></p> <p><Element type="idMap" extdescription="Placeholder for the WBXML Id-Offset map"></p> <p><AttributeType name="pointers" dttype="string"/></p> <p><attribute type="pointers" required="yes"/></p> <p></Element type="idMap"></p> <p><Element type="inlineCode" dttype="string" content="textOnly"/></p> <p><AttributeType name="constraintType" dttype="enumeration" dtvalues="validity too basic"/></p> <p><Element type="Constraint" content="textOnly"></p> <p><attribute type="constraintType" required="yes"/></p> <p><element type="inlineCode" minOccurs="1" maxOccurs="1"/></p> <p></Element type="Constraint"></p> <p><AttributeType name="functionName" dttype="string"/></p> <p><AttributeType name="event" dttype="string"/></p> <p><AttributeType name="argName" dttype="string"/></p> <p><AttributeType name="argDefault"/></p> <p><Element type="Argument"></p> <p><attribute type="argName" required="yes"/></p> <p><attribute type="argDefault" required="no"/></p> <p></Element type="Argument"></p> <p><Element type="Function"></p> <p><attribute type="functionName" required="yes"/></p> <p><element type="Argument" minOccurs="1"/></p> <p></Element type="Function"></p> <p><Element type="FunctionDef"></p> <p><attribute type="functionName" required="yes"/></p> <p><element type="Argument" minOccurs="1"/></p> <p></Element type="FunctionDef"></p> <p><AttributeType name="table" dttype="int"/></p> <p><AttributeType name="refTable" dttype="kref" extrefType="Table" extdataBinding="table"/></p> <p><AttributeType name="idField" dttype="id"/></p> <p><AttributeType name="dsr" dttype="string"/></p> <p><AttributeType name="refField" dttype="kref" extrefType="Field"/></p> <p><AttributeType name="refFields" dttype="krefs" extrefType="Field"/></p> <p><AttributeType name="type" dttype="enumeration" dtvalues="string integer datetime data time fixedDecimal boolean"/></p> <p><AttributeType name="length" dttype="int"/></p> <p><AttributeType name="decimalLength" dttype="int"/></p> <p><AttributeType name="defaultLabel" dttype="string"/></p> <p><AttributeType name="constant" dttype="string"/></p> <p><AttributeType name="paramIndex" dttype="int"/></p> <p><AttributeType name="id" dttype="id"/></p> <p><AttributeType name="file" dttype="string" exttranslatable="1"/></p> <p><AttributeType name="importance" dttype="enumeration" dtvalues="mandatory high normal low"/></p> <p><!-- Bitmap and icon support --></p>
---	---


```

<AttributeType name='iconFile' dttype='string'
exttype='iconFile'
<AttributeType name='bitmapFile' dttype='string'
exttype='iconFile'

<!-- new: Sorting -->
<!-- Disabled for VLO
<AttributeType name='order' dttype='enumeration'
dtvalues='ascending|descending'
<Element type='SortKey' content='empty'
<Attribute type='refField' required='yes'
<Attribute type='order' required='yes'
</Element type>

<Element type='SortSpec' content='refOnly'
<Attribute type='id' required='yes'
<Attribute type='title' required='yes'
<Element type='SortKey' minOccurs='1' maxOccurs='1'
</Element type>

<Element type='Search' content='refOnly'
<Attribute type='refField' required='yes'
</Element type>

<!-- new: Hyperlinking
We might also specify hyperlinks by more complex
queries, or by Constraints
<AttributeType name='ownField' dttype='idref'
extrefType='Field' extdataBinding='field'
<AttributeType name='destField' dttype='idref'
extrefType='Field' extdataBinding='field'
<AttributeType name='destFileSet' dttype='idref'
extrefType='FileSet'
<Element type='HyperLink' content='refOnly'
<Attribute type='ownField' required='yes'
<Attribute type='destField' required='yes'
<Attribute type='destFileSet' required='yes'
<Element type='VoiceCaption' minOccurs='0'
maxOccurs='1'
</Element type>

<AttributeType name='displayField' dttype='idref'
extrefType='Field' extdataBinding='field'
<Element type='Indirect' content='empty'
<Attribute type='refField' required='yes'
<Attribute type='displayField' required='yes'
</Element type>

<!-- START DataModel schema -->
<Element type='ConstantSelector' content='empty'
<Attribute type='refField' required='yes'
<Attribute type='constant' required='yes'
</Element type>

<Element type='ParameterSelector' content='empty'
<Attribute type='refField' required='yes'
<Attribute type='paramIndex' required='yes'
</Element type>

<Element type='Checktable' content='refOnly'
<Attribute type='refTable' required='yes'
<Attribute type='refField' required='yes'
<group order='many'
<element type='ConstantSelector' minOccurs='0'
<element type='ParameterSelector' minOccurs='0'
</group>
</Element type>

<!-- Model the relationships directly -->
<Element type='Relation' content='refOnly'
<Attribute type='table1' dttype='idref'
extrefType='Table'
<Attribute type='table2' dttype='idref'
extrefType='Table'
<Attribute type='field1' dttype='idref'
extrefType='Field'
<Attribute type='field2' dttype='idref'
extrefType='Field'
<Attribute type='cardinality1' dttype='enumeration'
dtvalues='1 n'
<Attribute type='cardinality2' dttype='enumeration'
dtvalues='1 n'

```

```

<AttributeType name='dependentObjects'
dttype='enumeration' dtvalues='table1 table2 neither'
default='neither'
<Attribute type='id' required='yes'
<Attribute type='table1' required='yes'
<Attribute type='table2' required='yes'
<Attribute type='field1' required='yes'
<Attribute type='field2' required='yes'
<Attribute type='cardinality1' required='yes'
<Attribute type='cardinality2' required='yes'
<Attribute type='dependentObjects' required='no'
</Element type>

<Element type='parentRelation' dttype='idref'
extrefType='Relation MainFile SubFile'
extrefScope='FileSet'

<!-- Linking: Another possibility is to use constraints.
A SubFile's ownField and parentField are replaced by an
attribute link which references the constraint.
The FK constraint is used in both directions, depending on
context.
<AttributeType name='baseTable' dttype='idref'
extrefType='Table'
<AttributeType name='baseField' dttype='idref'
extrefType='Field'
<AttributeType name='referringTable' dttype='idref'
extrefType='Table'
<AttributeType name='referringField' dttype='idref'
extrefType='Field'
<Element type='Constraint_ForeignKey'
content='refOnly'
<Attribute type='id' required='yes'
<Attribute type='baseTable' required='yes'
<Attribute type='baseField' required='yes'
<Attribute type='referringTable' required='yes'
<Attribute type='referringField' required='yes'
</Element type>

<AttributeType name='link' dttype='idref'
extrefType='Constraint_ForeignKey'
-->

<AttributeType name='value' dttype='string'
<Element type='ValueListElement' content='empty'
<Attribute type='value' required='yes'
</Element type>

<Element type='ValueList' content='refOnly'
<element type='ValueListElement' minOccurs='1'
maxOccurs='1'
</Element type>

<Element type='ValueRange' content='empty'
<Attribute type='lowerBound' dttype='string'
<Attribute type='lowerBoundInclusive'
dttype='boolean' default='1'
<Attribute type='upperBound' dttype='string'
<Attribute type='upperBoundInclusive'
dttype='boolean' default='1'
<Attribute type='lowerBound' required='no'
extdescription='Not present means not bound'
<Attribute type='lowerBoundInclusive' required='no'
<Attribute type='upperBound' required='no'
extdescription='Not present means not bound'
<Attribute type='upperBoundInclusive' required='no'
</Element type>

<Element type='DomainCheck' order='one'
content='refOnly'
<element type='Checktable' minOccurs='0'
maxOccurs='1'
<element type='ValueList' minOccurs='0' maxOccurs='1'
<element type='ValueRange' minOccurs='0'
maxOccurs='1'
</Element type>

<Element type='Field' content='refOnly'
<Attribute type='idField' required='yes'
<Attribute type='type' required='yes'
<Attribute type='length' required='no'
<Attribute type='maxLength' required='no'
<Attribute type='defaultLabel' required='no'
<Attribute type='nullAllowed' dttype='boolean'
<Attribute type='autoGenerate' dttype='enumeration' dtvalues='no integer GUID'
<Attribute type='nullAllowed' required='no' default='no'
<Attribute type='autoGenerate' required='no' default='no'

```

```

<element type="DomainCheck" minOccurs="0"
maxOccurs="1"/>
</Element>

<Element type="Index" content="empty">
<attribute type="relField" required="yes"/>
</Element>

<AttributeType name="externalKey" datatype="idref"
extrefType="Field" extrefScope="Table"/>
<Element type="Table" content="allOnly">
<attribute type="idTable" required="yes"/>
<attribute type="den" required="yes"/>
<attribute type="externalKey" required="no"/>
<element type="Field" minOccurs="1" maxOccurs="1"/>
<element type="Index" minOccurs="0"/>
</Element>

<Element type="DataModel" content="allOnly"
extrefType="Table" extrefScope="Table"/>
<group order="many" minOccurs="1">
<element type="Table"/>
<element type="Relation"/>
</group>
</Element>
<!-- END DataModel schema -->

<!--
Currently supported:
- Checkables, in the data model

To do:
- Complete modelling of controls
- Add support for popup frames

- Add Workflow support (see other design documentation):
Rules of form: if CNF(primitive constraints) then
setOf(Field, Value) + SetOf(Variable, Value)
Primitive constraint: (field, operator, value/field)

- Tradeoff between VBScript and builtin functionality is:
- builtin is declarative -> can be understood and
manipulated
- complexity of program vs general VBScript approach
- simplicity of customization vs. discontinuity

To Review
- Support for scripting
=> keep simple, to ease implementation, and also
validation (but reduces error-checking!)
- return value has fixed name in callee
- script won't support types, so pointers to indicate in
grammar
- Use Perl-like calling scheme, where formal
parameters have fixed names.
=> need to work out requirements a bit better

Notes on IE5 support:
- ?enumeration type not supported
- model 'open' is default
-->

<!--[CDATA[
One way of handling Complex fields:
<Complex label="Address">
<Group compress="yes" separator="<br>">
<ComplexSub datafield="Addr1"/>
<Break/>
<ComplexSub datafield="Addr2"/>
<Break/>
<Group compress="yes" separator="<br>">
<ComplexSub datafield="City"/>
<ComplexSub datafield="Region"/>
</Group>
</Group>
</Complex>
]]>

<AttributeType name="idFile" datatype="id"/>
<AttributeType name="text" datatype="string"/>
<AttributeType name="refFile" datatype="idref"
extrefType="File"/>

```

```

<AttributeType name="datafield" datatype="idref"
extrefType="Field" extrefScope="Table"/>
<AttributeType name="parentField" datatype="idref"
extrefType="Field"/>
<AttributeType name="sortBy" datatype="idref"
extrefType="SortSpec"/>
<AttributeType name="multilines" datatype="boolean"/>
<AttributeType name="navigationBar" datatype="boolean"/>

<AttributeType name="label" datatype="string"/>
<AttributeType name="visible" datatype="boolean"/>
<AttributeType name="navigate" datatype="boolean"/>
<AttributeType name="readonly" datatype="boolean"/>
<AttributeType name="mandatory" datatype="boolean"/>

<!-- For element Row -->
<AttributeType name="col" datatype="int" extrefType="all"/>
<AttributeType name="colspan" datatype="int"
extrefType="all"/>
<AttributeType name="align" datatype="enumeration"
dvalues="left center right" extrefType="all"/>
<AttributeType name="hspace" datatype="int"
extrefType="all"/>

<!-- For element GridPage -->
<AttributeType name="wrap" datatype="boolean"
extrefType="all"/>
<AttributeType name="cols" datatype="int" extrefType="all"/>

<Element type="LayoutAttr" content="allOnly"
extrefType="all">
<attribute type="col"/>
<attribute type="colspan"/>
<attribute type="align"/>
<attribute type="hspace"/>
</Element>

<Element type="GridPage" content="allOnly"
extrefType="all">
<attribute type="cols"/>
<attribute type="wrap" required="yes" default="1"/>
<element type="Row" minOccurs="0" maxOccurs="1"/>
</Element>

<!-- Code -->
<Element type="EventHandler" content="allOnly">
<attribute type="event"/>
<group order="any">
<element type="InlineCode"/>
<element type="FunctionCall"/>
</group>
</Element>

<AttributeType name="format" datatype="string"
extrefType="all"/>
<AttributeType name="fieldList" datatype="idref"
extrefType="Field"/>
<Element type="Caption">
<attribute type="format" required="yes"/>
<attribute type="fieldList" required="no"/>
</Element>
<Element type="VoiceCaption">
<attribute type="format" required="yes"/>
<attribute type="fieldList" required="yes"/>
</Element>

<!-- Tilesets, Tiles, etc. -->
<AttributeType name="targetFrame" datatype="enumeration"
dvalues="1 2 3 4 34 1234"/>
<AttributeType name="displayGroup" datatype="int"/>
<Element type="SubTitle">
<attribute type="title" required="yes"/>
<attribute type="id" required="no"/>
<attribute type="refFile" required="yes"/>
<attribute type="targetFrame" required="yes"/>
<attribute type="displayGroup" required="no"/>
<attribute type="parentRelation" required="yes"/>
<attribute type="sortBy" required="no"/>
<attribute type="visible" required="no" default="1"/>
<attribute type="importance" required="no"
default="normal"/>
<element type="Caption" minOccurs="0"
maxOccurs="1"/>
<element type="VoiceCaption" minOccurs="0"
maxOccurs="1"/>
<element type="SubTitle"/>

```

```

</ElementType>
<ElementType name="MainTitle">
  <attribute type="title" required="yes"/>
  <attribute type="id" required="no"/>
  <attribute type="refTitle" required="yes"/>
  <attribute type="targetFrame" required="yes"/>
  <attribute type="displayGroup" required="no"/>
  <attribute type="sortBy" required="no"/>
  <element type="Caption" minOccurs="0" maxOccurs="1"
    ext:description="Overrides the attribute"/>
  <element type="VoiceCaption" minOccurs="0"
    maxOccurs="1" ext:description="Overrides the attribute"/>
  <element type="SubTitle" maxOccurs="1"/>
</ElementType>
<ElementType name="DataControl">
  <attribute type="datafield" required="yes"/>
</ElementType>
<AttributeType name="displayCharacters" dt:type="int">
  <ElementType name="Column">
    <attribute type="title" required="yes"/>
    <attribute type="id" required="yes"/>
    <attribute type="datafield" required="yes"/>
    <attribute type="importance" required="no"
      default="normal"/>
    <attribute type="displayCharacters" required="no"/>
    <element type="Indirect" minOccurs="0" maxOccurs="1"/>
    <element type="VoiceCaption"/>
  </ElementType>
</Controls>
<!-- Next are advisory, may be overridden by Layout or
Engine components -->
<AttributeType name="minDisplayCharacters"
  dt:type="int">
  <ElementType name="Label" content="text"
    ext:control="1">
    <!-- Extends(DataControl) -->
    <attribute type="text" required="yes"/>
  </ElementType>
  <ElementType name="TextBlock" content="textOnly"
    ext:control="1">
    <!-- Extends(DataControl) -->
    <attribute type="label" required="no"/>
    <attribute type="id" required="yes"/>
    <attribute type="datafield" required="yes"/>
    <attribute type="minDisplayCharacters" required="no"/>
    <!-- attribute type="type" required="no" default="string"/>
    <!-- WHAT IS THIS FOR?? -->
    <attribute type="importance" required="no"
      default="normal"/>
    <attribute type="readonly" required="no" default="0"/>
    <attribute type="mandatory" required="no" default="0"/>
    <attribute type="multiline" required="no" default="0"/>
    <element type="HyperLink" minOccurs="0"
      maxOccurs="1"/>
    <element type="Indirect" minOccurs="0" maxOccurs="1"/>
    <element type="LayoutAttr" minOccurs="0"
      maxOccurs="1"/>
    <element type="EventHandler" minOccurs="0"/>
    <ext:SupportsType fieldType="string"/>
    <ext:SupportsType fieldType="Integer"/>
    <ext:SupportsType fieldType="float"/>
  </ElementType>
  <ElementType name="ComboBox" content="textOnly"
    ext:control="1">
    <!-- Extends(DataControl) -->
    <attribute type="label" required="no"/>
    <attribute type="id" required="yes"/>
    <attribute type="datafield" required="yes"/>
    <attribute type="minDisplayCharacters" required="no"/>
    <AttributeType name="showDescription"
      dt:type="boolean"/>
    <attribute type="showDescription" required="no"
      default="0"/>
    <!-- attribute type="type" required="no" default="string"/>
    <!-- WHAT IS THIS FOR?? -->
    <attribute type="importance" required="no"
      default="normal"/>
    <element type="HyperLink" minOccurs="0"
      maxOccurs="1"/>
    <element type="Indirect" minOccurs="0" maxOccurs="1"/>
    <element type="LayoutAttr" minOccurs="0"
      maxOccurs="1"/>
  </ElementType>

```

```

<element type="EventHandler" minOccurs="0"/>
<ext:SupportsType fieldType="string"/>
<ext:SupportsType fieldType="Integer"/>
</ElementType>
<!-- OptionGroups and Checkboxes -->
<ElementType name="Option" content="textOnly"
  ext:control="1">
  <attribute type="text" required="yes"/>
  <attribute type="value" required="yes"/>
  <attribute type="refTitle" required="no" ext:description="The
  title to show Page on. None means the parent of the Option
  control"/>
  <element type="Page" minOccurs="0" maxOccurs="1"
    ext:description="Page to show on this option being
    selected"/>
</ElementType>
<AttributeType name="default" dt:type="string">
<ElementType name="OptionGroup" ext:control="1">
  <attribute type="title" required="no"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <attribute type="readonly" required="no" default="0"/>
  <attribute type="minDisplayCharacters" required="no"/>
  <attribute type="default" required="no"/>
  <element type="Option" minOccurs="1" maxOccurs="1"/>
  <ext:SupportsType fieldType="string"/>
  <ext:SupportsType fieldType="Integer"/>
</ElementType>
<ElementType name="OnTrue">
  <attribute type="refTitle" required="yes"/>
  <attribute type="visible" required="no" default="0"/>
  <attribute type="navigate" required="no"/>
</ElementType>
<ElementType name="OnFalse">
  <attribute type="refTitle" required="yes"/>
  <attribute type="visible" required="no" default="0"/>
  <attribute type="navigate" required="no"/>
</ElementType>
<ElementType name="CheckBox" content="textOnly"
  ext:control="1">
  <!-- Extends(DataControl) -->
  <attribute type="text" required="yes"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <attribute type="readonly" required="no" default="0"/>
  <attribute type="minDisplayCharacters" required="no"/>
  <element type="OnFalse" minOccurs="0" maxOccurs="1"/>
  <element type="OnTrue" minOccurs="0" maxOccurs="1"/>
  <element type="HyperLink" minOccurs="0"
    maxOccurs="1"/>
  <element type="LayoutAttr" minOccurs="0"
    maxOccurs="1"/>
  <element type="EventHandler" minOccurs="0"/>
  <ext:SupportsType fieldType="boolean"/>
</ElementType>
<ElementType name="Memo" content="textOnly"
  ext:control="1">
  <attribute type="label" required="no"/>
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>
  <attribute type="importance" required="no"
    default="normal"/>
  <attribute type="readonly" required="no" default="0"/>
  <attribute type="minDisplayCharacters" required="no"/>
  <attribute type="mandatory" required="no" default="0"/>
  <!-- Extends(DataControl) -->
  <element type="Indirect" minOccurs="0" maxOccurs="1"/>
  <element type="LayoutAttr" minOccurs="0"
    maxOccurs="1"/>
  <element type="EventHandler" minOccurs="0"/>
  <ext:SupportsType fieldType="string"/>
</ElementType>
<ElementType name="DateTime" content="textOnly"
  ext:control="1">
  <!-- Extends(DataControl) -->
  <attribute type="id" required="yes"/>
  <attribute type="datafield" required="yes"/>

```

```

<attribute type='importance' required='no'
default='normal'/>
<attribute type='readonly' required='no' default='0'/>
<attribute type='mandatory' required='no' default='0'/>
<attribute type='label' required='no'/>
<attribute type='minDisplayCharacters' required='no'/>
<element type='HyperLink' minOccurs='0'
maxOccurs='1'/>
<element type='LayoutAtr' minOccurs='0'
maxOccurs='1'/>
<ext:SupportsType fieldType='date'/>
<ext:SupportsType fieldType='time'/>
<ext:SupportsType fieldType='datetime'/>
</ElementType>

<!-- Special controls: Address -->
<AttributeType name='addr1' dtype='string'/>
<AttributeType name='addr2' dtype='string'/>
<AttributeType name='city' dtype='string'/>
<AttributeType name='region' dtype='string'/>
<AttributeType name='country' dtype='string'/>
<AttributeType name='postal' dtype='string'/>

<ElementType name='Address' content='eltOnly'
ext:control='1'>
<!-- # Extends(DataControl) -->
<attribute type='id' required='yes'/>
<attribute type='importance' required='no'
default='normal'/>
<attribute type='readonly' required='no' default='0'/>
<attribute type='mandatory' required='no' default='0'/>
<attribute type='minDisplayCharacters' required='no'/>
<attribute type='label' required='no'/>

<attribute type='addr1' required='no'/>
<attribute type='addr2' required='no'/>
<attribute type='city' required='no'/>
<attribute type='region' required='no'/>
<attribute type='country' required='no'/>
<attribute type='postal' required='no'/>

<element type='EventHandler' minOccurs='0'/>
</ElementType>

<!-- should MonthlyCalendar & DaySchedule be controls
or Tiles? -->
<ElementType name='MonthlyCalendar' content='eltOnly'>
<attribute type='id' required='yes'/>
<ext:SupportsType fieldType='date'/>
<ext:SupportsType fieldType='datetime'/>
</ElementType>

<ElementType name='DaySchedule' content='eltOnly'>
<attribute type='id' required='yes'/>
<attribute type='datefield' required='yes'/>
<ext:SupportsType fieldType='date'/>
<ext:SupportsType fieldType='datetime'/>
</ElementType>

<AttributeType name='displayFormat' dtype='string'/>
<ElementType name='ComplexField'>
<attribute type='id' required='yes'/>
<attribute type='datefield' required='yes'/>
<attribute type='type' required='no' default='string'/>
<!-- # Extends(Tile) -->
<attribute type='importance' required='no'
default='normal'/>
<attribute type='readonly' required='no' default='0'/>
<attribute type='mandatory' required='no' default='0'/>
<attribute type='minDisplayCharacters' required='no'/>

<attribute type='displayFormat' required='yes'/>
<group minOccurs='1' maxOccurs='1' order='many'>
<!-- element type=DataControl -->
<element type='Address'/>
<element type='DateTime'/>
<element type='CheckBox'/>
<element type='ComboBox'/>
<element type='ComplexField'/>
<element type='Memo'/>
<element type='OptionGroup'/>
<element type='TextBox'/>
</group>
</ElementType>

```

```

<!-- Commands -->
<AttributeType name='command' dtype='enumeration'
defaultValue='faxdoc telephone'/>
<ElementType name='StandardCommand'>
<attribute type='command' required='yes'/>
<element type='Argument' minOccurs='1'/>
</ElementType>
<ElementType name='AddInCommand'>
<attribute type='functionName' required='yes'/>
<element type='Argument' minOccurs='1'/>
</ElementType>

<AttributeType name='topMenu' dtype='string'/>
<ElementType name='MenuItem' content='eltOnly'>
<attribute type='label' required='yes'/>
<attribute type='topMenu' required='yes'/>
<group order='none'>
<element type='StandardCommand'/>
<element type='AddInCommand'/>
</group>
</ElementType>

<!-- Shareable elements -->
<ElementType name='Row' content='eltOnly'
ext:displayUnit='mergable' ext:layout='all'>
<group minOccurs='0' maxOccurs='1' order='many'>
<!-- element type=DataControl -->
<element type='Address'/>
<element type='DateTime'/>
<element type='CheckBox'/>
<element type='ComboBox'/>
<element type='ComplexField'/>
<element type='Memo'/>
<element type='OptionGroup'/>
<element type='TextBox'/>
</group>
</ElementType>

<ElementType name='Page' content='eltOnly'
ext:displayUnit='mergable' ext:layout='all'>
<attribute type='id'/>
<attribute type='file' required='yes'
ext:description='Used if the Page is not merged'/>
<group minOccurs='0' maxOccurs='1' order='many'>
<!-- element type=DataControl -->
<element type='Address'/>
<element type='DateTime'/>
<element type='CheckBox'/>
<element type='ComboBox'/>
<element type='ComplexField'/>
<element type='Memo'/>
<element type='OptionGroup'/>
<element type='TextBox'/>
</group>
</ElementType>

<ElementType name='Tile' content='eltOnly'
ext:displayUnit='unmergable' ext:fragment='1'>
<attribute type='idTile' required='yes'/>
<attribute type='refTable' required='yes'/>
</ElementType>

<ElementType name='FrameTile' content='eltOnly'
ext:displayUnit='unmergable' ext:fragment='1'>
<attribute type='idTile' required='yes'/>
<!-- # Extends(Tile) -->
<attribute type='refTable' required='yes'/>
<group minOccurs='1' order='many'>
<element type='Page'/>
<element type='GridPage'/>
<element type='EventHandler'/>
</group>
</ElementType>

<AttributeType name='defaultSearchField' dtype='idref'
ext:refType='Field'/>
<ElementType name='ListViewTile' content='eltOnly'
ext:displayUnit='unmergable' ext:fragment='1'>
<attribute type='idTile' required='yes'/>
<attribute type='refTable' required='yes'/>
<!-- # attribute type=defaultSearchField required='no'/>
-->
<!-- element type=Search, minOccurs='0'
maxOccurs='1'/>
<!-- element type=HyperLink, minOccurs='0'
maxOccurs='1'/>
<!-- element type=VoiceCaption, minOccurs='0'
maxOccurs='1' ext:description='Describes the row'/>

```

```

<- element type="SortSpec" minOccurs="0"
maxOccurs="1" ->
<-#Extends(Tile)->
<group minOccurs="1" maxOccurs="1">
  <element type="Column"/>
</group>
</Element type>

<- *** these two Field setting tags assume to the
MainTile table we are manipulating *** ->
<- Copy from (current table?) to field in new object ->
<ElementType name="CopyField" content="eltOnly">
  <attribute type="destField" required="yes"/>
  <attribute type="srcField" required="yes"/>
</Element type>

<- Set field in new object ->
<ElementType name="FieldValue" content="eltOnly">
  <attribute type="destField" required="yes"/>
  <attribute type="value" required="yes"/>
</Element type>

<AttributeType name="refModify" dtype="tstr"
exts:display="Tile"/>
<ElementType name="OnNow" content="eltOnly">
  <attribute type="destTileSet" required="yes"/>
  <element type="CopyField" minOccurs="0"
maxOccurs="1"/>
  <element type="FieldValue" minOccurs="0"
maxOccurs="1"/>
</Element type>

<ElementType name="OnCopy" content="eltOnly">
  <attribute type="destTileSet" required="yes"/>
  <element type="CopyField" minOccurs="1"
maxOccurs="1"/>
  <element type="FieldValue" minOccurs="0"
maxOccurs="1"/>
</Element type>

<ElementType name="TileSet" content="eltOnly"
exts:displayUnit="unmergeable">
  <attribute type="tile" required="yes"/>
  <attribute type="id" required="yes"/>
  <attribute type="refModify" required="no"/>
  <attribute type="navigationBar" required="no"
default="0"/>
  <attribute type="readonly" required="no" default="1"/>
  <attribute type="tmapFile" required="no"/>
  <element type="OnNow" minOccurs="0"
maxOccurs="1"/>
  <element type="OnCopy" minOccurs="0"
maxOccurs="1"/>
  <element type="MainTile" minOccurs="1"
maxOccurs="1"/>
</Element type>

<ElementType name="TileSet" content="eltOnly">
  <group minOccurs="1">
    <element type="TileSet"/>
  </group>
</Element type>

<- Shareable element pools ->
<ElementType name="Tables" content="eltOnly">
  <element type="Table" minOccurs="1"/>
</Element type>

<ElementType name="Tiles" content="eltOnly">
  <group minOccurs="1" order="many">
    <element type="ListViewTile"/>
    <element type="FrameTile"/>
  </group>
</Element type>

<ElementType name="Functions" content="eltOnly">
  <group minOccurs="0">
    <element type="Function"/>
  </group>
</Element type>

<- Middleware Information ->
<AttributeType name="scenarioName" dtype="string"
ext:minLength="3" ext:maxLength="10"/>
<AttributeType name="structureId" dtype="string"
ext:length="3"/>
<ElementType name="ScenarioSegment">
  <attribute type="structureId" required="yes"/>
  <attribute type="refTable" required="yes"/>
</Element type>

```

```

<ElementType name="Scenario">
  <attribute type="scenarioName" required="yes"/>
  <attribute type="id" required="no"/>
  <element type="ScenarioSegment" minOccurs="1"
maxOccurs="1"/>
</Element type>

<ElementType name="Middleware">
  <element type="Scenario" minOccurs="1"
maxOccurs="1"/>
</Element type>

<- Top level ->
<ElementType name="GUI" content="eltOnly">
  <attribute type="tile" required="yes"/>
  <attribute type="iconFile" required="no"/>
  <element type="DataModel" minOccurs="1"
maxOccurs="1"/>
  <element type="Middleware" minOccurs="0"
maxOccurs="1"/>
  <element type="Tile" minOccurs="1" maxOccurs="1"/>
  <element type="Functions" minOccurs="0"
maxOccurs="1"/>
  <element type="TileSet" minOccurs="0"
maxOccurs="1"/>
</Element type>

</Schema>

```

8.2 MDF Workbench

The Workbench for MDF provides context-sensitive guidance to the developer driven from the MDF language definition (an XML schema). In the middle of FIG. 4 you can see a popup window responding to the developer's "insert node" command by displaying a list of candidate elements. The list of candidates is determined by the Workbench by inspecting the MDF language definition.

In the bottom of FIG. 4, you can see a Preview of what the application will look like on one of the supported layout themes – in this case, the Handheld PC. The Workbench provides screens that approximate the look and behavior of the Themes in order to help the application developer visualize what the application will look like when deployed.

8.3 MDF Interpreter

The generic interpreter is described in section 6.3.

Step five, the realization of the generic UI Model, of the Application System creation process described in section 6.1 can be performed by use of an object programming language such as C++. Similarly, step six, the realization of the MDF Themes can be performed by use of said programming language.

8.3.1 MDF Layout Themes

In this section, we illustrate the independence of the UI Model from the device-specific screen layouts ("Themes"). A single MDF application definition is shown with three graphical user-interface Themes: Half-VGA Handheld PC, Palm PC, and VGA Handheld PC, applicable to the Windows CE (Microsoft Corp, Seattle, WA) devices with corresponding screen sizes.

1.1 Handheld PC (half-VGA size)

The half-VGA Handheld PC (240 pixels high by 640 pixels wide) screen layout is illustrated in FIG.9. The mapping of the generic UI model to the Theme is as follows:

- (a) the titles of the top-level Tile-Set assemblies are displayed in the form of a Navigation Bar on the upper-left;
- (b) the Navigation Stack is exposed as a list on the lower-left. Clicking on an entry will remove from the stack all entries deeper than the selected entry. Additionally, a backward-pointing arrow will remove the topmost element from the stack;
- (c) ListView Tiles and Frame Tile Textbox elements may have associated Hyperlinks. Clicking on a Hyperlink invokes it, creating a new Tilesset, which is added to the top of the Navigation Stack. The old current Tilesset is hidden and the new one displayed.
- (d) the Main Tile of the current Tilesset is shown immediately to the right of the Navigation Bar in the middle window, and the Sub tiles are shown in a Tabstrip at the right;

1.2 Palm PC

The Palm PC (320 pixels high by 240 pixels wide) screen layout is the smallest GUI layout supported by the MDF prototype. FIG.10 illustrates the visual appearance. In contrast to the Handheld PC theme, the list of top-level Tilesets is accessed through the Menu, instead of by a Navigation Bar. The Navigation Stack is not exposed as a visible control although as in the Half-VGA Handheld PC Theme it can be manipulated via the back arrow on the toolbar. The Main Tile of the current Tilesset is shown as the leftmost tab in the Tabstrip, and the Sub Tiles follow to the right.

1.3 Handheld PC (full-VGA size)

The full VGA (640 pixels high by 480 pixels wide) screen layout allows more information to be displayed on the screen at the same time. FIG.11 illustrates the visual appearance. We therefore move away from the use of a Tabstrip to organize the Tiles, and instead display the Tiles simultaneously in one or more fixed locations or "slots" on the screen.

In detail, the mapping of the generic UI model to the full-VGA Theme is similar to that of the Half-VGA Theme; steps (a) to (c) are the same.

Step (d) is replaced as follows:

the screen is split into 4 areas called slots;

an additional attribute, the `targetFrame`, is defined on the Main Tile and Sub Tile of each Tileset. This attribute specifies the set of slots which the Tile should occupy on the screen;

on initial display of a Tile Set, the Tiles of the Tile set are examined in order. If the slots specified in the `targetFrame` attribute for the Tile are unoccupied, the Tile is displayed in those slots, otherwise it is not displayed at all;

Tiles that do not occupy any of the slots are displayed in minimized form as captions in the overflow areas at the bottom part of the screen. By clicking on one of this captions, the user forces display of the associated Tile at the slots specified in its `targetFrame` attribute, and removes the associated caption from the overflow area;

when a Tile is to be displayed because of said user action, the screen slots it is to occupy are removed from the Tile or Tiles that are currently displayed;

if such a Tile no longer has the minimum set of slots it needs, it releases all of its slots and displays itself as a caption in the overflow area;

otherwise, said Tile reduces its displayed size to fit the slots it is currently allocated;

the UI elements displayed on a Tile may vary depending on the number of slots allocated to the Tile. For example, the columns of a ListView Tile have an associated importance attribute that is used to determine which columns to suppress in order to fit the available space. The illustration of FIG.11 shows the upper Customer List reduced to half width in response to the co-opting of the right half of its slot allocation by the Customer

Detail Tile.

The following is Table2 which is a simple Service Management demonstration program, defined with the MDF application language.

Table2

```

<GUI xmlns="x-schema:ApplicationSchema.xml" title="Test
GUI Definition">
  <DataModel>
    <!-- Code Tables -->
    <Table IdTable="t:SN_Status" dsn="SN_Status">
      <Field IdField="t:SN_Status.Code" type="string"/>
      <Field IdField="t:SN_Status.Text" type="string"/>
    </Table>
    <Table IdTable="t:Equip_Type" dsn="Equip_Type">
      <Field IdField="t:Equip_Type.Code" type="string"/>
      <Field IdField="t:Equip_Type.Text" type="string"/>
    </Table>
    <Table IdTable="t:WR_Act" dsn="WR_Act">
      <Field IdField="t:WR_Act.Code" type="string"/>
      <Field IdField="t:WR_Act.Text" type="string"/>
    </Table>
    <!-- BEGIN: TABLES for synchronization -->
    <Table IdTable="t:SN" dsn="SN">
      <Field IdField="t:SN.NoteNum" defaultLabel="Request
      Id" type="string" length="8"/>
      <Field IdField="t:SN.Priority" defaultLabel="Priority"
      type="string" length="1"/>
      <Field IdField="t:SN.NoteDate"
      defaultLabel="Received" type="date" length="8"/>
      <Field IdField="t:SN.CustNum"
      defaultLabel="Customer Id" type="string" length="8"/>
      <Field IdField="t:SN.SerialNum"
      defaultLabel="Installation Id" type="string" length="18"/>
      <Field IdField="t:SN.EquipmentNum"
      defaultLabel="Equipment Number" type="string" length="18"/>
      <Field IdField="t:SN.ShortText"
      defaultLabel="Description" type="string" length="20"/>
      <Field IdField="t:SN.LongText" defaultLabel="Memo"
      type="string" length="80"/>
      <Field IdField="t:SN.Status" defaultLabel="Completed"
      type="string" length="1"/>
      <Field IdField="t:SN.ContactName"
      defaultLabel="Contact" type="string" length="30"/>
      <!-- Not in Jason's model -->
      <Field IdField="t:SN.ModelNo" defaultLabel="Model"
      type="string" length="20"/>
    </Table>
    <Table IdTable="t:Equip" dsn="Equip">
      <Field IdField="t:Equip.EquipmentNum"
      defaultLabel="Equipment Number" type="string" length="18"/>
      <Field IdField="t:Equip.InstallDate"
      defaultLabel="Installed" type="date" length="8"/>
      <Field IdField="t:Equip.EquipmentType"
      defaultLabel="Equipment Type" type="string" length="1"/>
      <Field IdField="t:Equip.ModelNo"
      defaultLabel="Model" type="string" length="20"/>
      <Field IdField="t:Equip.CustNum"
      defaultLabel="Customer Id" type="string" length="8"/>
      <!-- Not in Jason's model. Subsumed by
      EquipmentNum -->
      <Field IdField="t:Equip.Serial" defaultLabel="Serial No"
      type="integer" length="18"/>
    </Table>
    <Table IdTable="t:Cust" dsn="Cust">
      <Field IdField="t:Cust.CustNum"
      defaultLabel="Customer Id" type="string" length="8"/>
      <Field IdField="t:Cust.Name" defaultLabel="Name"
      type="string" length="15"/>
      <Field IdField="t:Cust.City" defaultLabel="City"
      type="string" length="10"/>
      <Field IdField="t:Cust.PostalCode" defaultLabel="Post"
      type="string" length="10"/>
      <!-- should add validation code here -->
      <Field IdField="t:Cust.Telephone" defaultLabel="Tel"
      type="string" length="15"/>
      <!-- Not in Jason's model, but required -->
      <Field IdField="t:Cust.Addr1" defaultLabel="Addr1"
      type="string" length="15"/>
      <Field IdField="t:Cust.Addr2" defaultLabel="Addr2"
      type="string" length="15"/>
      <Field IdField="t:Cust.Region" defaultLabel="Region"
      type="string" length="15"/>
      <index refField="t:Cust.Name"/>
      <index refField="t:Cust.City"/>
      <index refField="t:Cust.PostalCode"/>
    </Table>
    <!-- WorkReport: check different from t:WorkReport, add
    relationships as required -->
    <Table IdTable="t:WorkReport" dsn="WorkReport">
      <Field IdField="t:WR.Id" defaultLabel="Report Id"
      type="string" length="15"/>
      <Field IdField="t:WR.NoteNum"
      defaultLabel="Notification" type="string" length="8"/>
      <Field IdField="t:WR.ActNum"
      defaultLabel="Activity" type="string" length="2"/>
      <Field IdField="t:WR.ActCode" defaultLabel="Activity
      Code" type="string" length="4"/>
      <Field IdField="t:WR.ActTime" defaultLabel="Memo"
      type="string" length="80"/>
      <Field IdField="t:WR.ActDate" defaultLabel="Date"
      type="date" length="8"/>
      <Field IdField="t:WR.ActTime" defaultLabel="Time"
      type="time" length="8"/>
      <Field IdField="t:WR.RequiredPart" defaultLabel="Part
      No" type="string" length="10"/>
      <Field IdField="t:WR.PartDelivery"
      defaultLabel="Delivery date" type="date" length="8"/>
      <Field IdField="t:WR.EquipmentNum"
      defaultLabel="Equipment Id" type="string" length="18"/>
      <Field IdField="t:WR.CustNum"
      defaultLabel="Customer Id" type="string" length="8"/>
    </Table>
    <!-- END: TABLES for synchronization -->
    <Table IdTable="t:Model" dsn="Model">
      <Field IdField="t:Model.ModelNo" defaultLabel="Model
      No." type="string" length="20"/>
      <Field IdField="t:Model.Manufacturer"
      defaultLabel="Manufacturer" type="string" length="15"/>
      <Field IdField="t:Model.Description"
      defaultLabel="Description" type="string" length="15"/>
    </Table>
    <Table IdTable="t:Symp" dsn="Symp">
      <Field IdField="t:Symp.Id" defaultLabel="Id"
      type="string" length="27"/>
      <Field IdField="t:Symp.Description"
      defaultLabel="Description" type="string" length="20"/>
      <Field IdField="t:Symp.ModelNo" defaultLabel="Model"
      type="string" length="20"/>
    </Table>
    <Table IdTable="t:Check" dsn="Check">
      <Field IdField="t:Check.Id" defaultLabel="Id"
      type="string" length="3"/>
      <Field IdField="t:Check.Likelihood"
      defaultLabel="Likelihood" type="string" length="8"/>
      <Field IdField="t:Check.Description"
      defaultLabel="Description" type="string" length="15"/>
    </Table>
    <Table IdTable="t:Part" dsn="Part">
      <Field IdField="t:Part.MatNo" defaultLabel="Material"
      type="string" length="10"/>
      <Field IdField="t:Part.Text" defaultLabel="Description"
      type="string" length="18"/>
    </Table>
    <!-- Relations -->
    <Relation Id="rein:Cust_SN" table1="t:Cust"
    field1="t:Cust.CustNum" table2="t:SN" field2="t:SN.CustNum"
    cardinality1="1" cardinality2="n"/>
    <Relation Id="rein:WR_SN" table1="t:WorkReport"
    field1="t:WR.NoteNum" table2="t:SN" field2="t:SN.NoteNum"
    cardinality1="n" cardinality2="1"/>
    <Relation Id="rein:Model_Symp" table1="t:Model"
    field1="t:Model.ModelNo" table2="t:Symp"
    field2="t:Symp.ModelNo" cardinality1="1" cardinality2="n"/>
    <Relation Id="rein:Cust_Equip" table1="t:Equip"
    field1="t:Equip.CustNum" table2="t:Cust"
    field2="t:Cust.CustNum" cardinality1="n" cardinality2="1"/>
  </DataModel>

```

```

<Relation id="rein:WR_Equip" table1="t:WorkReport"
field1="t:WR.EquipmentNum" table2="t:Equip"
field2="t:Equip.EquipmentNum" cardinality1="1"
cardinality2="1"/>
<Relation id="rein:SN_Equip" table1="t:SN"
field1="t:SN.EquipmentNum" table2="t:Equip"
field2="t:Equip.EquipmentNum" cardinality1="1"
cardinality2="1"/>
<Relation id="rein:WR_Cust" table1="t:WorkReport"
field1="t:WR.CustNum" table2="t:Cust"
field2="t:Cust.CustNum" cardinality1="1" cardinality2="1"/>
<Relation id="rein:Equip_Model" table1="t:Equip"
table2="t:Model" field1="t:Equip.ModelNo"
field2="t:Model.ModelNo" cardinality1="1" cardinality2="1"/>
<Relation id="rein:WR_ActCode" table1="t:WorkReport"
table2="t:WR_ActCode" field1="t:WR.ActCode"
field2="t:WR.ActCode" cardinality1="1" cardinality2="1"/>
<DataModel>
<MiddleWare>
<Scenario scenarioName="Customer" id="1d1">
<ScenarioSegment structureId="TOP" refTable="t:Cust?"/>
<ScenarioSegment structureId="010" refTable="t:Equip?"/>
</Scenario>
<Scenario scenarioName="ServiceNotification" id="1d2">
<ScenarioSegment structureId="TOP" refTable="t:SN?"/>
</Scenario>
</MiddleWare>
</DataModel>
<FrameTitle idTitle="SN_General" refTable="t:SN">
<Page title="General">
<TextBox id="c00000" datafield="t:Cust.CustNum"
label="Requester">
<HyperLink ownField="t:Cust.CustNum"
destField="t:Cust.CustNum" destTitleSet="ts01F"/>
</TextBox>
<TextBox id="c00001" datafield="t:SN.CustNum"
label="CustomerRef"/>
<TextBox id="c00002" datafield="t:SN.ModelNo"
label="ModelRef"/>
<TextBox id="c00003" datafield="t:SN.SerialNum"
label="EquipmentRef"/>
<CheckBox id="c00004" datafield="t:SN.Status"
text="Completed/Show completion tab">
<OnTrue refTable="ts01F.Report" visible="1"
navigate="1"/>
</CheckBox>
</Page>
<Page title="Memo">
<Memo id="c01000" datafield="t:SN.LongText"/>
</Page>
</FrameTitle>
<FrameTitle idTitle="Cust_General" refTable="t:Cust">
<Page title="General">
<TextBox id="c11000" datafield="t:Cust.CustNum"
label="CustomerRef"/>
<TextBox id="c11001" datafield="t:Cust.Name"
label="NameRef"/>
<TextBox id="c11002" datafield="t:Cust.Telephone"
readonly="0" mandatory="0" multiline="0"/>
<TextBox id="c11003" datafield="t:Cust.Addr1"
readonly="0" mandatory="0" multiline="0"/>
<TextBox id="c11004" datafield="t:Cust.Addr2"
readonly="0" mandatory="0" multiline="0"/>
<TextBox id="c11005" datafield="t:Cust.Region"
readonly="0" mandatory="0" multiline="0"/>
<TextBox id="c11006" datafield="t:Cust.PostalCode"
label="Postal"/>
</Page>
</FrameTitle>
<ListViewTitle idTitle="Cust_List" refTable="t:Cust">
<HyperLink ownField="t:Cust.CustNum"
destField="t:Cust.CustNum" destTitleSet="ts00F"/>
<Column id="c20000" title="ID"
datafield="t:Cust.CustNum"/>
<Column id="c20001" title="Name"
datafield="t:Cust.Name"/>
<Column id="c20002" title="Area"
datafield="t:Cust.City"/>
</ListViewTitle>
<ListViewTitle idTitle="Cust_SN_List" refTable="t:SN">
<HyperLink ownField="t:SN.NoteNum"
destField="t:SN.NoteNum" destTitleSet="ts01F"/>
<Column id="c40000" title="T"
datafield="t:SN.Priority"/>
<Column id="c40001" title="Date"
datafield="t:SN.NoteDate"/>
<Column id="c40002" title="Description"
datafield="t:SN.ShortText"/>
</ListViewTitle>
<ListViewTitle idTitle="SN_MainDetail"
refTable="t:WorkReport">
<HyperLink ownField="t:WR.Id" destField="t:WR.Id"
destTitleSet="ts06F">
<Column id="c50000" title="Service Date"
datafield="t:WR.ActDate"/>
<Column id="c50001" title="Description"
datafield="t:WR.ActText"/>
</ListViewTitle>
<FrameTitle idTitle="Report" refTable="t:WorkReport">
<Page title="Header" id="qh001">
</Page>
<GridPage wrap="1" cols="2">
<Row>
<TextBox id="TextBox_Dec7193511_200"
datafield="t:WR.LF" readonly="0" mandatory="0" multiline="0"
label="MFR"/>
</Row>
<Row>
<DateTime id="DateTime_Dec4135008_210"
datafield="t:WR.ActDate" readonly="0" mandatory="0"
label="ServiceDate"/>
</Row>
<Row>
<DateTime id="DateTime_Dec4135028_220"
datafield="t:WR.ActTime" readonly="0" mandatory="0"
label="Time"/>
</Row>
<GridPage>
<Page title="Report" id="qh002">

```

```

</TitleSet>
<TitleSet id="ts:00F" title="Customer">
  <MainTitle refTitle="Cust_General" title="Customer User"
targetFrame="1">
    <SubTitle refTitle="v:Equipment" title="Equip."
parentRelation="rein:Cust_Equip" targetFrame="3">
        </SubTitle>
    <SubTitle refTitle="Cust_SN_User" title="Note"
parentRelation="rein:Cust_SN" targetFrame="4">
        </SubTitle>
    <SubTitle refTitle="Cust_MaintHist" title="Hist"
parentRelation="rein:WR_Cust" targetFrame="4">
        </SubTitle>
    </MainTitle>
</TitleSet>
<TitleSet id="ts:01" title="Nothostions"
navigationBase="1">
  <MainTitle refTitle="ts:01.Main" refTitle="SN_User"
title="Request" targetFrame="1">
    <SubTitle refTitle="SN_General" title="General"
parentRelation="ts:01.Main" targetFrame="2">
        </SubTitle>
    <SubTitle refTitle="Cust_General" title="Cust"
parentRelation="rein:Cust_SN" targetFrame="2">
        </SubTitle>
    <SubTitle refTitle="SN_MaintHist" title="Hist"
parentRelation="rein:WR_SN" targetFrame="3">
        </SubTitle>
    </MainTitle>
</TitleSet>
<TitleSet id="ts:01F" title="Notification"
navigationBase="0">
  <MainTitle refTitle="SN_General" title="General"
targetFrame="1">
    <SubTitle refTitle="Cust_General" title="Cust"
parentRelation="rein:Cust_SN" targetFrame="2">
        </SubTitle>
    <SubTitle refTitle="SN_MaintHist" title="Hist"
parentRelation="rein:WR_SN" targetFrame="3">
        </SubTitle>
    <SubTitle refTitle="ts:01F.Report" refTitle="Report"
title="Report" parentRelation="rein:WR_SN" visible="N"
targetFrame="2">
        </SubTitle>
    </MainTitle>
</TitleSet>
<TitleSet id="ts:02" title="Installation" navigationBase="1">
  <MainTitle refTitle="ts:02.Main" refTitle="v:Equipment"
title="Installation" targetFrame="1">
    <SubTitle refTitle="Installation" title="General"
parentRelation="ts:02.Main" targetFrame="2">
        </SubTitle>
    <SubTitle refTitle="SN_MaintHist" title="Hist"
parentRelation="rein:WR_Equip" targetFrame="3/4">
        </SubTitle>
    </MainTitle>
</TitleSet>
<TitleSet id="ts:02F" title="Installation"
navigationBase="1">
  <MainTitle refTitle="Installation"
title="Installation" targetFrame="12">
    <SubTitle refTitle="SN_MaintHist" title="Hist"
parentRelation="rein:WR_Equip" targetFrame="3/4">
        </SubTitle>
    </MainTitle>
</TitleSet>
<TitleSet id="ts:06F" title="Maintenance Record"
navigationBase="1">
  <MainTitle refTitle="Maintenance" title="Record"
targetFrame="123/4">
    </MainTitle>
</TitleSet>
</TitleSet>
</OBJ>

```

4. Brief description of the drawings

FIG. 1 is an overview of the process for creating an application.

FIG. 2 illustrates the range of deployment configurations for an application.

FIG. 3 is an overview of the Workbench architecture.

FIG. 4 is a view of a computer screen showing the Workbench screen layout.

FIG. 5 is a view of the Workbench during insertion of a new element instance.

FIG. 6 is a view of the Workbench during the setting of an attribute value.

FIG. 7 is an overview of the Interpreter architecture.

FIG. 8 is a detail of the operation of the Parsing Services component of the interpreter.

FIG. 9 is a series of views of a computer screen illustrating the 240*640 pixel Handheld PC layout Theme for the MDF application of Table 2.

FIG. 10 is a series of views of a computer screen illustrating the Palm PC layout Theme for the MDF application of Table 2.

FIG. 11 is a series of views of a computer screen illustrating the 480*640 pixel Handheld PC layout Theme for the MDF application of Table 2.

FIG. 12 is a series of views of the display of a microbrowser-enabled wireless telephone illustrating the server-side Interpreter's WAP Theme for the MDF application of Table 2.

FIG. 13 is a transcript of a dialog conducted over a telephone between a user and the server-side Interpreter's Voice Theme executing the MDF application of Table 2.

FIG. 1.

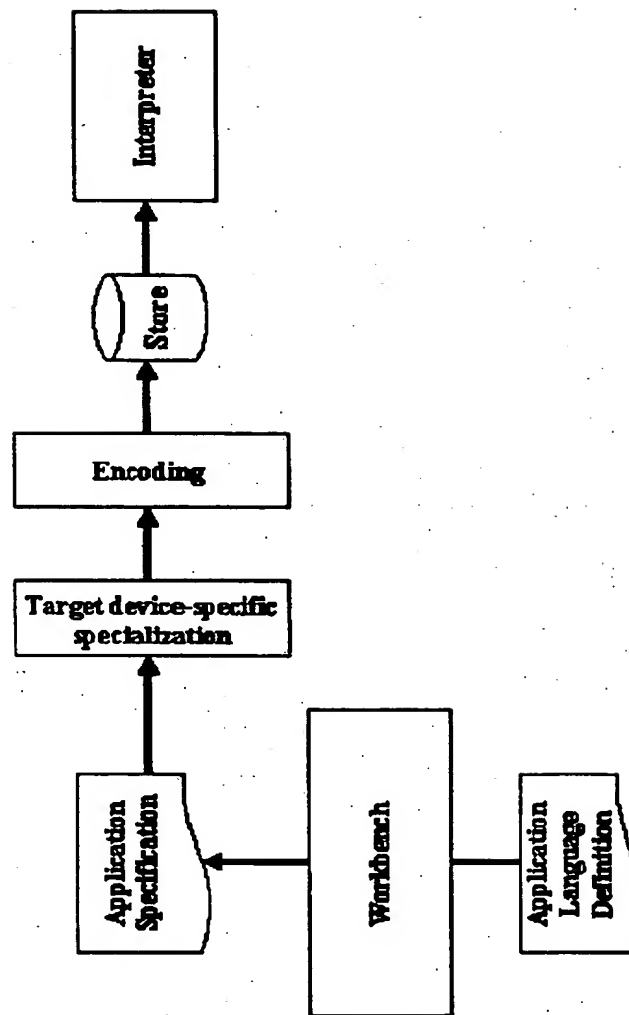


FIG. 2

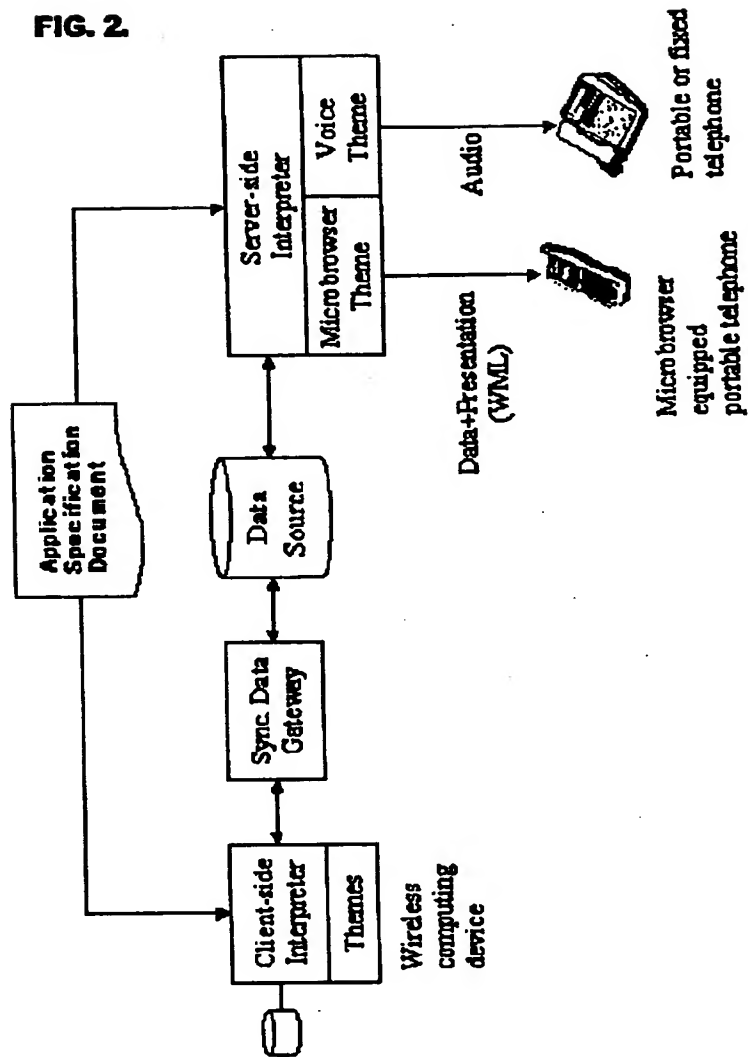


FIG. 3.

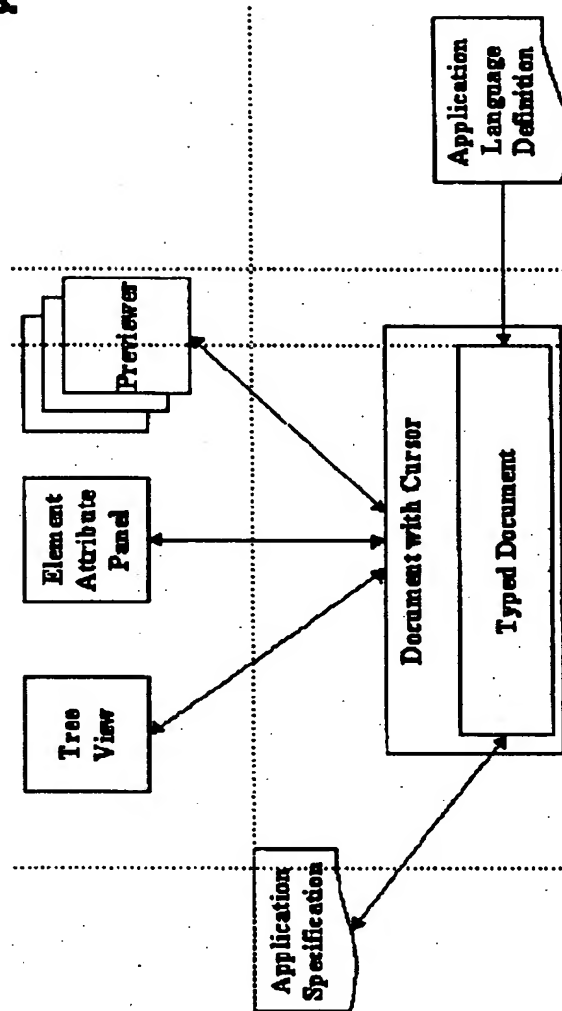


FIG. 4.

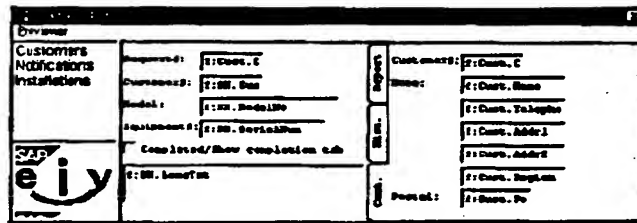
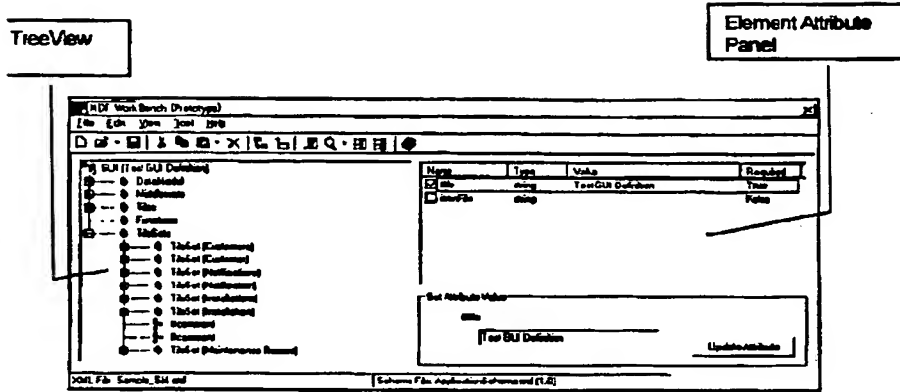
Figure 4a -
Handheld PC Previewer

Figure 4a - Palm PC Previewer

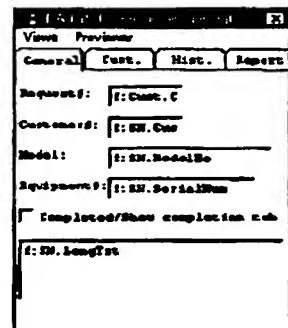


Figure 4a - VGA size Handheld PC Previewer

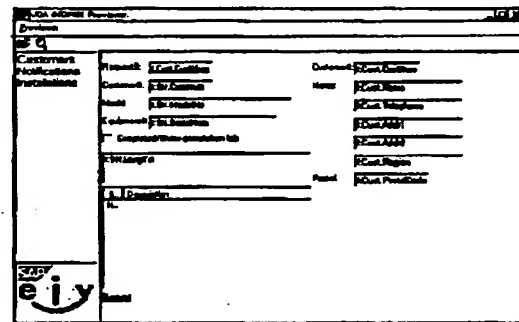
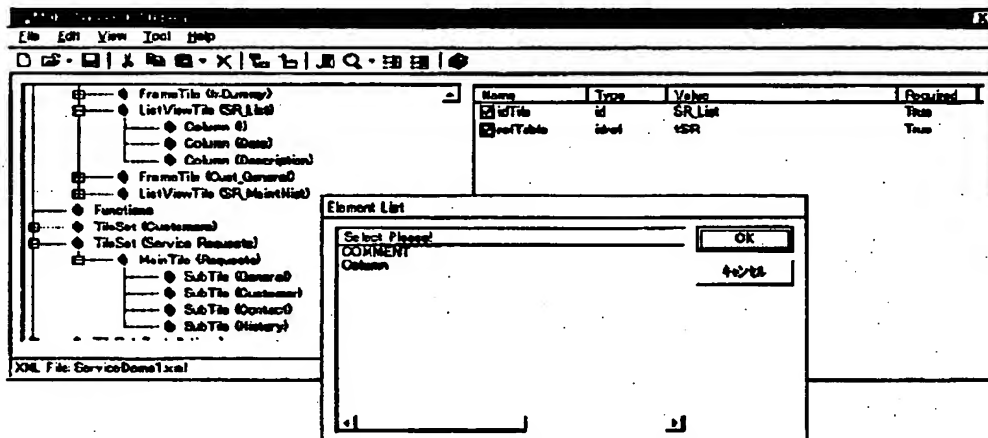
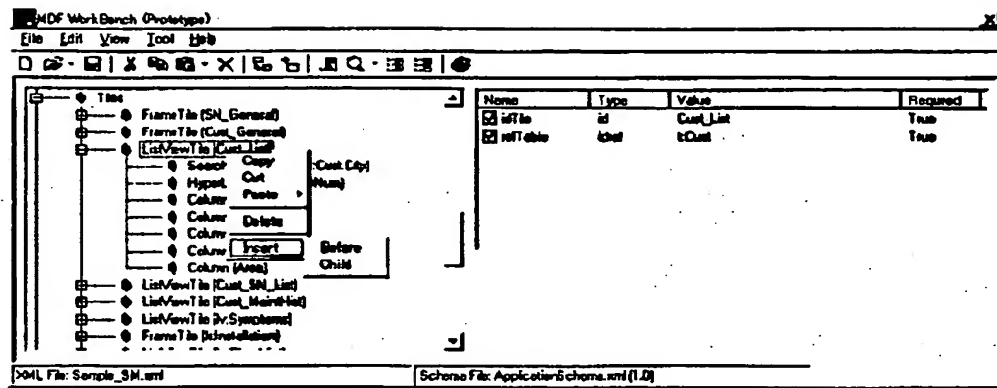


FIG. 5.



```
<AttributeType name='cardinality1' dt:type='enumeration' dt:values='1 n'/>
```

[illegible]

FIG. 7.

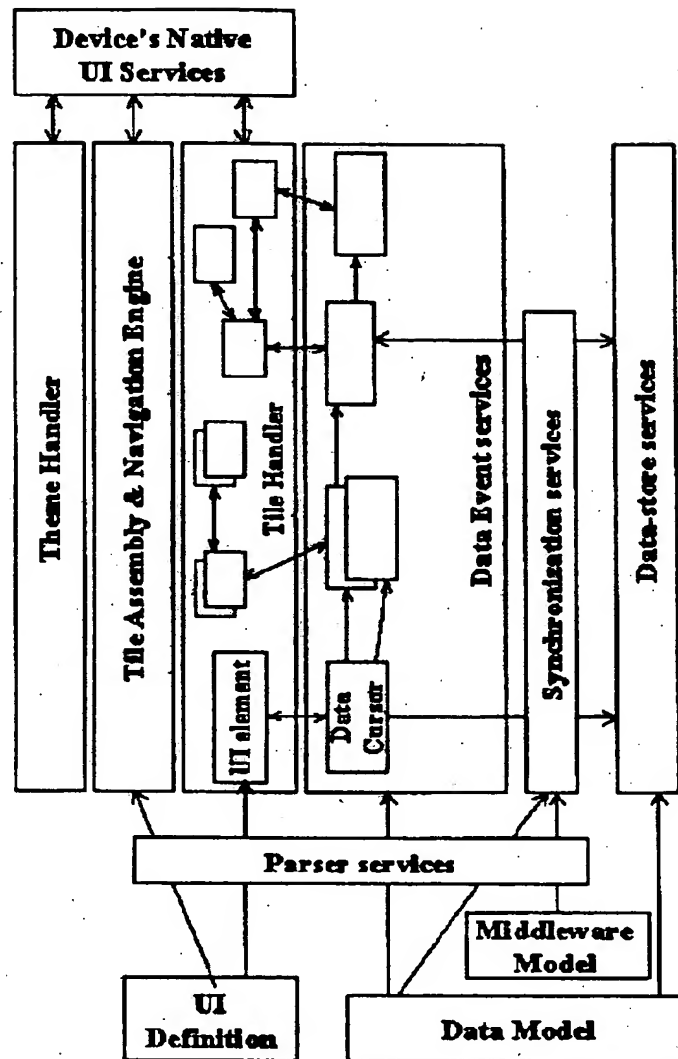


FIG. 8.

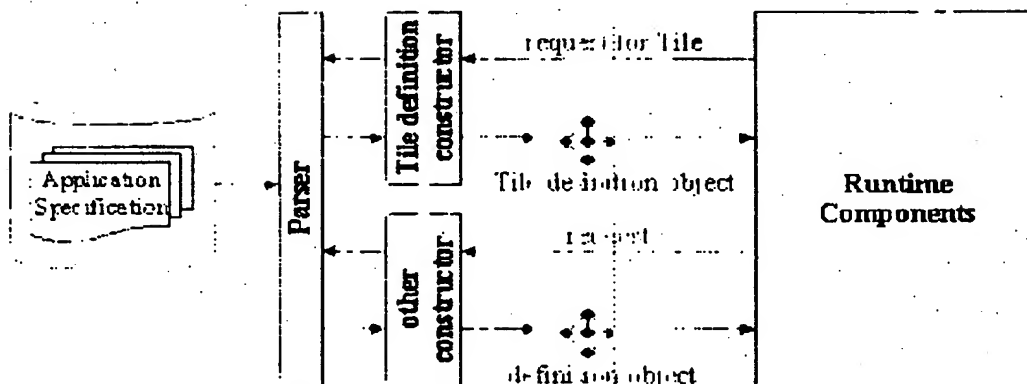


FIG. 9.

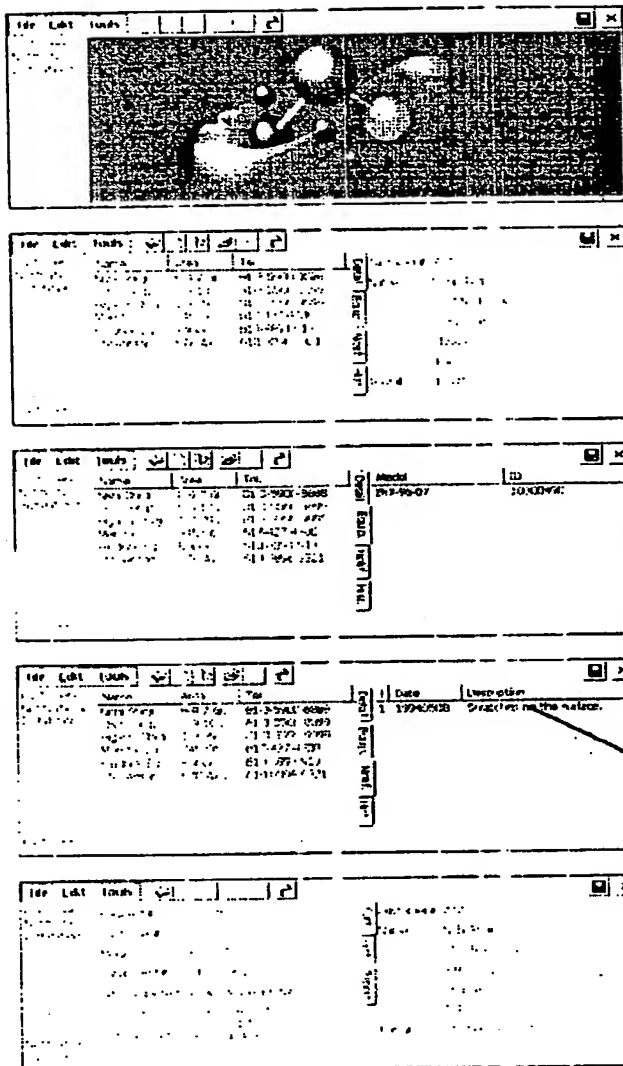


FIG. 10.

Top Left Screenshot: The 'Customers' tab is active. The left sidebar shows a tree view with 'Customers', 'Notifications', 'Installations', 'Specs', and 'Edit'. The main area is empty.

Top Middle Screenshot: The 'Customers' tab is active. The main area displays a list of customers with columns 'Name' and 'Area'.

Name	Area
Nishi Shop	9-6-7 Giza Ch...
Old Computer Base...	5-6-10 Ruten...
Miyoshi Shop	1-3-30 Nishi-Op...
Mori Co. Ltd.	245 Congress...
Pradick Equipment	Rocky Hiller R...
Staverson Co. Ltd.	676 Avenue of...

Top Right Screenshot: The 'Customers' tab is active. The main area displays customer details for Customer # 777.
 Customer # 777
 Name: Nishi Shop
 Address: 9-6-7 Giza Ch...
 City: Giza
 Postal: 11-8880

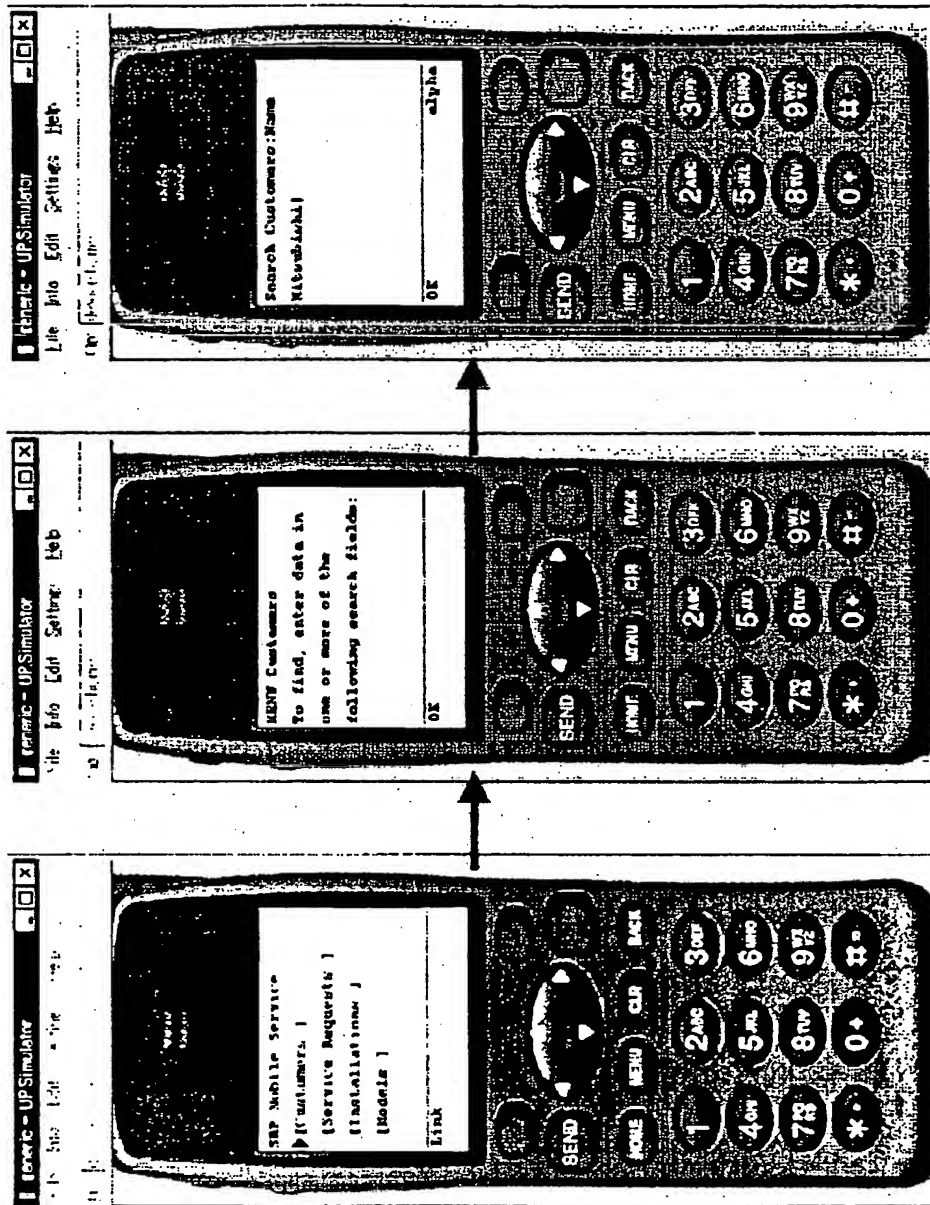
Bottom Left Screenshot: The 'Customers' tab is active. The main area displays a list of notifications with columns 'Date' and 'Description'.

Date	Description
199403...	Scratches on the surface.

Bottom Right Screenshot: The 'Customers' tab is active. The main area displays a detailed view of a notification.
 Request # 30000816
 Customer # 777
 Model: 284-96-07
 Equipment #: 123-007880123-0888
☒ Completed/Show completion tab
 Scratches on surface. Does not cause serious problems right now but can lead to quick repairs.

[illegible][illegible][illegible]

FIG. 12.



Abstract

The present invention relates generally to the creation of data-centric applications for computing devices. The invention is particularly characterized for defining classes of data-centric applications that share particular generic user interface models comprising the following steps; determining the types of distinguished User Interface elements, and their constituent elements, required for said generic User Interface Model; determining the cardinality and relationships between the Tiles in groups of Tiles called Assemblies; determining the logical model for Navigation between the Assemblies, and the required user operations on the model state.

BLANK PAGE

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

BLANK PAGE